Word Reordering for Statistical Machine Translation via Modeling Structural Differences between Languages

Isao Goto

March 2014

Kyoto University

Abstract

An increase in the globalization of various fields has highlighted the need for machine translation. Current machine translation research is focused on statistical machine translation (SMT). Machine translation involves two central tasks: *word selection* and *reordering*. Current SMT methods work well for word selection because word selection can often be solved using local information, such as contextual information contained in a phrase and n-grams. Therefore, SMT works well for translation between languages with similar word orders. However, current SMT methods do not work well for long-distance word reordering between languages with largely different word orders. This is because neither phrase-level local information nor n-grams are sufficient for long-distance word reordering. Thus, the translation quality of SMT between languages with largely different word orders is lower than that between languages with similar word orders.

In machine translation, modeling the structural differences between a source language and a target language facilitates the process of reordering. Such models can be used to estimate word order in the target language from a sentence in the source language. A straightforward way to model structural differences between languages is parsing sentences in both source and target languages. This process enables identification of the structural differences. However, there are many languages in the world, and high-quality parsers are available for a small number of languages. Thus, there is a great need for SMT methods that do not require a source language parser and/or a target language parser. The objective of this thesis is to propose ways to model structural differences between languages for improved reordering in SMT, given certain restrictions regarding the availability of parsers.

Chapter 1 introduces the history of machine translation research. We then describe current issues in statistical machine translation research, state the objective of the thesis, and provide an overview of our approaches.

Chapter 2 introduces the SMT framework, reordering methods, and evaluation methods.

Chapter 3 proposes a new distortion model for phrase-based SMT. The model is based on word sequence labeling, and calculates probabilities of reordering without requiring a parser. The proposed model uses label sequences to approximately model structural differences between languages. Our model can learn the effect of relative word order among candidate words to be translated next. In addition, our model can learn the effect of distances from the training data.

Chapter 4 proposes a post-ordering method that reorders target words based on syntactic structures for Japanese-to-English translation using a target language parser. The existing post-ordering method reorders a sequence of target language words in a word order similar to that of the source language, resulting in a target language word order via phrase-based SMT. In our method, the sequence is reordered by (1) parsing the sequence using inversion transduction grammar (ITG) to obtain syntactic structures that are similar to those in the source language, and (2) transferring the obtained syntactic structures into target language syntactic structures according to the ITG.

Chapter 5 proposes a pre-ordering method that reorders source language sentences using a target language parser without a source language parser. To train the ITG parsing model that uses syntactic categories to parse source language sentences, we produce source language syntactic structures by: (1) projecting the constituent structures of sentences in the target language to corresponding sentences in the source language, (2) producing probabilistic models for parsing using the projected partial structures and the Pitman-Yor process, and (3) producing full binary syntactic structures within the constraints of the projected partial structures by parsing using the probabilistic models.

Chapter 6 compares the three proposed reordering methods.

Chapter 7 concludes this thesis and describes future work.

Acknowledgments

I would like to express my sincere appreciation to my supervisor, Professor Sadao Kurohashi for his continuous encouragement and guidance during my thesis research.

I am also grateful to my thesis committee members: Professor Katsumi Tanaka and Professor Tatsuya Kawahara of Kyoto University, for their valuable suggestions and comments about my thesis research.

I would like to express my gratitude to Dr. Masao Utiyama of NICT, for providing me with generous advice and assistance.

I am grateful to Dr. Eiichiro Sumita for helpful suggestions and for offering many research opportunities at NICT.

I would like to thank the members of NICT. I would especially like to thank the members of the multilingual translation laboratory for their helpful suggestions, Dr. Kiyotaka Uchimoto for advice and assistance when I joined NICT, and Jewel Faulkner, who improved the quality of the English in my papers.

I am grateful to the members of the natural language processing group at the NHK Science & Technology Research Laboratories for their helpful suggestions and encouragement. I would especially like to thank Dr. Hideki Tanaka, who gave me suggestions for improving my thesis and encouraged me during this work, Dr. Terumasa Ehara, who supported me during the early years of my natural language processing research at NHK, and Dr. Ichiro Yamada, who encouraged me when we started working for NICT.

I would also like to thank the members of the Language Media Laboratory at Kyoto University for their helpful suggestions.

Finally, I would like to thank my wife, children, and parents for their contin-

uous support and encouragement during and before this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Machine Translation Research

Recent developments in communication technology and transportation have contributed to the increasing globalization of various fields. As a result, there is a growing demand for ways to conduct international exchanges of information, such as understanding information written in foreign languages and transmitting information in multiple languages. This situation has led to a tremendous need for machine translation, in which text is automatically translated between languages.

Machine translation research began soon after the invention of computers (in the latter half of the 1940s). The first approach to machine translation was rule-based where the rules of translation knowledge were generated by humans. Rule-based machine translation research was actively conducted until the 1990s. However, it became clear that producing rules by human had inherent limitations in treating complex linguistic phenomena. To overcome these limitations, researchers proposed several data-driven approaches, in which translation knowledge is automatically acquired from a bilingual corpus. In 1981, a framework for machine translation by analogy, which was the basis of example-based machine translation, was proposed by Nagao [1984]. Around 1990, a method called statistical machine translation (SMT) was proposed by Brown et al. [1993]. In SMT, translation knowledge is treated probabilistically, and probabilistic translation knowledge is automatically produced from a large-scale bilingual corpus. Around

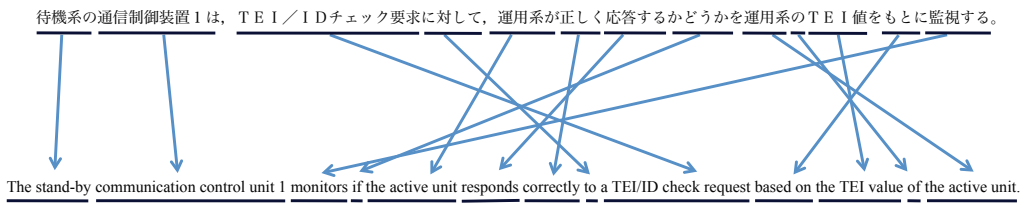

Figure 1.1: Example of a Japanese-English parallel sentence pair.

2000, improvements in computer processing power and the increased availability of large-scale bilingual corpora and fundamental software led to an intensification of SMT research. Since then, the SMT approach has been studied actively worldwide.

1.2 Current Issues in SMT

Machine translation comprises two main tasks: *word selection* and *reordering*. Early SMT [Brown et al., 1993] used single words as translation units. However, the size of this unit is problematic in that contextual information is not efficiently addressed. To solve this problem, phrase-based SMT [Koehn et al., 2003a], which uses a sequence of words (a phrase) as a translation unit, was proposed. When phrases are used as translation units, contextual information in phrases can be retained and used efficiently. This improved the quality of word selection because word selection is often solved using phrase-level local information or n-grams. Therefore, current SMT methods actually work well for translating between languages with similar word orders and for translating short sentences, such as travel conversation. However, phrase-level local context information is not sufficient for long distance word reordering between languages with largely different word orders. Thus, the translation quality of current SMT between languages with largely different word orders is lower than that between languages with similar word orders. For example, Figure 1.1 shows an example of parallel Japanese (a subject-object-verb (SOV) language) and English (a subject-verb-object (SVO) language) sentence pairs. There are many complicated instances of long-distance word reordering between the Japanese sentence and the corresponding English

sentence. When only local context information is considered, it is difficult to conduct such complicated instances of long-distance word reordering.

1.3 Thesis Objective: Modeling Structural Differences between Languages for Reordering

Sentences must follow certain structural rules to make sense. This means that word order is restricted by the structures of individual languages (this is called syntax), and each language has a distinct syntax. Therefore, in translation, the word order that follows the target language syntax should be estimated from a source language sentence whose word order follows the source language syntax.

In machine translation, it is important to model differences in syntactic structures between a source language and a target language for reordering. This is because the model for differences in syntactic structures between languages can be used to estimate a target language word order from a source language sentence, and if the model can capture essential differences in syntactic structures, this will facilitate reordering. Modeling differences in syntactic structures is especially important for translating between languages with largely different word orders.

A straightforward way to model structural differences between languages is to parse sentences in both the source and target languages, thus capturing the structural differences. However, the resources that are available for an SMT system vary by language. There are many languages in the world with insufficient language resources. Specifically, the languages for which high-quality parsers are available are few. Therefore, in addition to SMT methods that use both a source language parser and a target language parser, there is a great need for SMT methods that work without a source language parser, a target language parser, or without either type of parser.

The objective of the thesis is as follows. Under certain restrictions associated with the availability of parsers, structural differences between languages can be modeled in ways that are better than existing modeling methods to improve reordering in SMT.

1.4 Overview of Our Approaches

As explained in the previous section, there are many languages for which a parser is not available. Therefore, to translate without a parser, an SMT method that does not require a parser is necessary.

Presently, many people are able to understand major languages in addition to their native languages and major languages are used for information sharing and international discussion. For example, English is usually used for international conferences and English and French are used as the working languages at the United Nations secretariat. To effectively address this situation, machine translation is needed from various languages into major languages. In many cases, major languages have rich language resources, including parsers. Thus, there is a great need for SMT methods that do not require a source language parser but use a target language parser.

Therefore, we propose methods for modeling structural differences between languages in the following two cases: (1) no parsers are required and (2) a target language parser is required.

(1) For cases in which no parsers are required, we propose a new distortion model, which estimates word reordering in phrase-based SMT, which is one SMT method in which word selection and reordering are jointly conducted. The proposed model can approximately model structural differences between languages using label sequences that can characterize elements of syntax, such as VP or NP, without a parser. (See Chapter 3 for more detail.)

(2) For cases in which a target language parser is available, we propose improved post-ordering and pre-ordering methods. In post-ordering methods, reordering is conducted after word selection and in pre-ordering methods, reordering is conducted before word selection. These methods, in which word selection and reordering are conducted separately, can estimate reordering by efficiently using syntactic structures.

    We propose a new post-ordering method that uses syntactic structures by parsing a target language word sequence in a word order similar to that of

the source language, whereas an existing post-ordering method does not use syntactic structures. (Chapter 4 describes this concept in detail.)

We also propose a new pre-ordering method that uses syntactic structures by projecting the syntactic structures of target language sentences onto the corresponding source language sentences to produce a parsing model for source language sentences. (Chapter 5 describes this concept in detail.)

1.5 Organization of the Thesis

The rest of this thesis is organized as follows.

Chapter 2 introduces the SMT framework, and describes reordering and evaluation methods.

Chapter 3 discusses distortion models for phrase-based statistical machine translation. We explain some issues with previous distortion models. We then describe the proposed distortion model based on word sequence labeling and present evaluation results.

Chapter 4 describes post-ordering for Japanese-English statistical machine translation. We introduce the post-ordering framework and explain previous research and associated issues. We then explain the proposed post-ordering methods by parsing and present evaluation results.

Chapter 5 discusses pre-ordering for statistical machine translation. We show the pre-ordering framework and explain previous work in terms of applicabilities. We then describe the proposed pre-ordering method using a target language parser and present evaluation results.

Chapter 6 compares the three proposed reordering methods.

Chapter 7 concludes the thesis and describes future work.

Chapter 2

Technical Introduction

2.1 Statistical Machine Translation Framework

Machine translation can be thought of as the problem of finding the most likely target language sentence E given a source language sentence F using a set of parameters θ

$$\hat{E} = \underset{E}{\operatorname{argmax}}\, P(E|F;\theta). \tag{2.1}$$

In the current SMT model, $P(\cdot)$ is assumed to be represented by a log-linear model that is a linear combination of the products of feature functions $f_i(E,F)$ and weight parameters w_i

$$\hat{E} = \underset{E}{\operatorname{argmax}} \sum_i w_i f_i(E,F). \tag{2.2}$$

Each feature function captures a different aspect of translation, such as the logarithm of a phrase translation model probability or the logarithm of a language model probability. A translation model calculates the probability of F given E or the probability of E given F. A language model calculates the probability of E without conditioning on F.

Reordering methods in SMT can be classified into the following three types, depending on the timing of word selection and reordering: (1) Joint-ordering, in which target word selection and reordering occur simultaneously, (2) Post-ordering, in which reordering takes place after word selection, and (3) Pre-ordering,

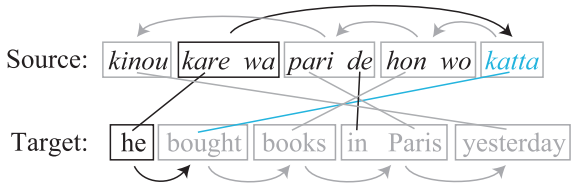

Figure 2.1: An example of left-to-right Japanese-English translation. Boxes represent phrases and arrows indicate the translation order of the phrases.

in which reordering occurs before word selection. We will explain these three types in the following text.

2.2 Joint-ordering

In joint-ordering, target word selection and reordering take place simultaneously. In this method, translation models are classified into two types: phrase-based and tree-based. The SMT method that uses the phrase-based translation model is called phrase-based SMT, while the SMT method that uses the tree-based translation model is called tree-based SMT. In phrase-based SMT, reordering is conducted using distortion models. In tree-based SMT, word selection and reordering are based on synchronous CFG models. We describe these two models in the subsequent sections.

2.2.1 Distortion Model

In the phrase-based SMT [Koehn et al., 2007; Koehn, 2010], the source language sentence F is broken up into I source phrases $\bar{f}_i$ and each source phrase $\bar{f}_i$ is translated into an target phrase $\bar{e}_i$. The probability of the translation model is calculated as $\prod_{i=1}^{I} P(\bar{f}_i|\bar{e}_i)$. This translation model does not capture reordering of phrases.

Therefore, a distortion model, which calculates probabilities of reordering of phrases, is necessary. In phrase-based SMT, target hypotheses are generated sequentially from left to right. Therefore, the role of the distortion model is to estimate the source phrase position to be translated next whose target side phrase

will be located immediately to the right of the already generated hypotheses, given the last translated source word position. An example is shown in Figure 2.1. In Figure 2.1, we assume that only the *kare wa* (English: "he") has been translated. The target word to be generated next will then be "bought" and the source word to be translated next will be the corresponding Japanese word *katta*. Thus, a distortion model should estimate phrases including *katta* as source phrase positions to be translated next. In the existing method described by [Koehn et al., 2007; Koehn, 2010], a distortion model calculates the probability of a start_i given an end_{i-1} and F, where start_i is defined as the first position of the source phrase that is translated to the i-th target phrase and end_{i-1} as the last position of the source phrase that is translated to the $(i-1)$-th target phrase.

A naive distortion model, also called a linear distortion cost model, considers only the distance between phrases. The probability of the linear distortion cost model is calculated as

$$\prod_{i=1}^{I} \alpha^{|\text{start}_i - \text{end}_{i-1} - 1|}, \alpha \in [0, 1]. \tag{2.3}$$

This distortion model penalizes long-distance reordering.

The linear distortion cost model does not consider words. However, the phrase to be translated next depends on the words in the source language sentence. Thus, it is helpful to use words in phrases when calculating the probabilities of reordering of phrases. Simple modeling of probability for each distance conditioned on phrase pairs will lead to data sparseness. There will be many combinations of a distance and a phrase pair that do not occur in a given training data set, especially for large distances. This makes it difficult to reliably estimate probability distributions using such frequencies.

The MSD lexicalized reordering model [Koehn et al., 2005] can be used to avoid this problem. In this model, probabilities are calculated for only three orientations: monotone, swap, and discontinuous. The probabilities for these three orientations are conditioned on only the last translated phrase or only a phrase candidate to be translated next. In the monotone orientation, the source phrase to be translated next is on the right, adjacent to the last translated phrase, in the swap orientation, the source phrase to be translated next is on the left,

adjacent to the last translated phrase, and the discontinuous orientation describes all other phrases; that is, all the phrases that are not adjacent to the last translated phrase.

2.2.2 Synchronous CFG and ITG

In the tree-based SMT, a synchronous context-free grammar (CFG) [Aho and Ullman, 1969; Aho and Ullman, 1972] is used for phrase reordering. A synchronous CFG is an extension of a context-free grammar, and consists of bilingual pairs of CFG rules. We will explain synchronous CFG using the following examples. A CFG rule for Japanese that defines a verb phrase (VP) as consisting of a noun phrase (NP) and a VP is as follows:

$$\text{VP} \rightarrow \text{NP VP}$$

A CFG rule for English that defines a verb phrase (VP) as consisting of a VP and an NP is as follows:

$$\text{VP} \rightarrow \text{VP NP}$$

A synchronous CFG rule that defines a bilingual CFG rule pair is as follows:

$$\text{VP} \rightarrow \text{NP}_1\ \text{VP}_2 \mid \text{VP}_2\ \text{NP}_1$$

This synchronous CFG rule maps a nonterminal symbol VP to a pair of symbol sequences. The left side of the "|" is associated with the source language (Japanese) and the right side of the "|" is associated with the target language (English). The indexes for the generated nonterminal symbols represent the correspondences of nonterminals between languages. This rule can capture the differences in the order of an NP and a VP in a VP in Japanese and English. Below are examples of synchronous CFG rules.[1]

$$\text{S} \rightarrow \text{NP}_1\ \text{VP}_2 \mid \text{NP}_1\ \text{VP}_2$$
$$\text{VP} \rightarrow \text{NP}_1\ \text{VP}_2 \mid \text{VP}_2\ \text{NP}_1$$

[1] Although PP is usually used as a phrase label for phrases including a post position, we use NP as a phrase label for phrases including a post position, *wa*, *ga*, or *wo* in Japanese, to fit Japanese phrase labels to English phrase labels.

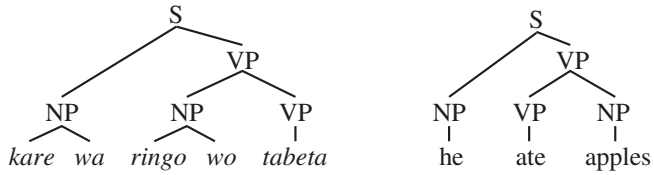

Figure 2.2: Example of bilingual syntactic structures.

NP $\rightarrow$ *kare wa* | he
NP $\rightarrow$ *ringo wo* | apples
VP $\rightarrow$ *tabeta* | ate

When we use these synchronous CFG rules to parse a Japanese sentence

kare wa ringo wo tabeta,

we are able to obtain the corresponding target language syntactic structure and sentence as well as the source language syntactic structure, as shown in Figure 2.2.

Synchronous CFG rules can address word selection and word reordering simultaneously. The translation model probabilities are the product of each probability of the rules used to parse an input sentence. When a rule is represented in the format $x \rightarrow \alpha \mid \alpha'$, the following probability distributions may be used as the probabilities of the rules. $P(\alpha, \alpha'|x)$ is the probability of the derivation of the source and target trees. $P(\alpha'|x, \alpha)$ is the probability of the derivation of the target tree, given the source tree.

When using syntax, syntactic categories are used for nonterminals. When syntax is not used for the source language, the target language, or either language, a non-syntactic nonterminal symbol, such as X, is used as the nonterminal symbol for the languages without syntax.

An inversion transduction grammar (ITG) [Wu, 1997] is a special case of a synchronous CFG. ITG allows only two reordering ways: the same order or the reverse order. Each ITG rule can be expressed by two symbols or less on the right-hand side of an arrow in a synchronous CFG rule for each language. When syntactic categories of ITG are not used and only one nonterminal symbol, such as X, is used, the grammar is called bracketing transduction grammar (BTG).

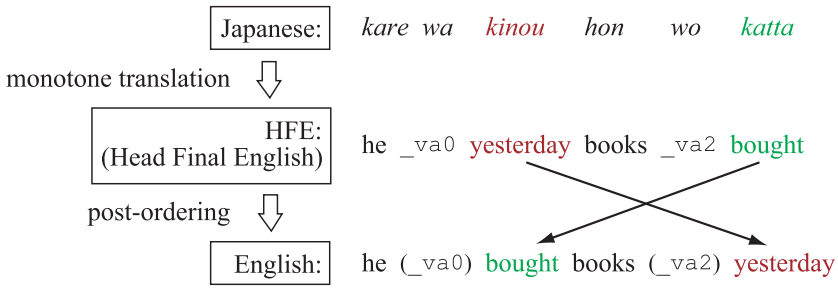

Figure 2.3: Post-ordering framework.

2.3 Post-ordering

A post-ordering approach proposed by [Sudoh et al., 2011b] for Japanese-English translation can be used to carry out translation as a two-step process involving word selection and reordering. The translation flow for the post-ordering method is shown in Figure 2.3, where "HFE" is an abbreviation of "Head Final English", which represents target language (English) words in almost the same word order as that of a source language (Japanese).[2] The two-step process is as follows.

1. Translating first almost monotonously transforms a source language (Japanese) sentence into an HFE sentence. This can be done using phrase-based SMT [Koehn et al., 2003b], which can produce accurate translations when only local reordering is required.

2. Reordering then transforms the HFE sentence into a target language (English) sentence. Sudoh et al. [2011b] proposed a reordering model that consisted of an HFE-English phrase-based SMT, which reordered by translating HFE sentences into English sentences.

Pre-ordering rules from a target language to a source language are needed to realize this framework because the pre-ordering rules are used to produce HFE training sentences from target language training sentences.

[2]Explanations of pseudo-particles (_va0 and _va2) and other details specific to HFE are given in Section 4.4.4.

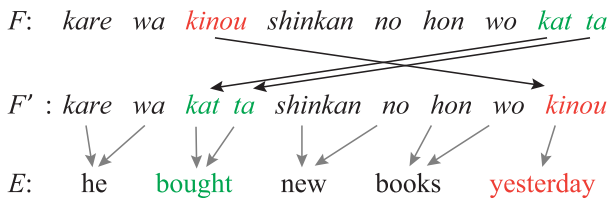

Figure 2.4: Pre-ordering framework (Japanese-English translation example).

2.4 Pre-ordering

For long-distance word reordering, the syntactic structure of a source language sentence F is useful. The pre-ordering approach is an SMT method that can simply use the syntactic structure of F. This approach performs translation as a two-step process, as shown in Figure 2.4. The two-step process is as follows.

1. The first process reorders F to F', which is a source language word sequence in almost the same word order as that of the target language.

2. The second process translates F' into a target language sentence E using an SMT method, such as phrase-based SMT, which can produce accurate translations when only local reordering is required.

In most pre-ordering research, word reordering is achived using reordering rules and the syntactic structure of F, obtained via a source language syntactic parser. Pre-ordering rules can be produced automatically [Xia and McCord, 2004; Genzel, 2010] or manually [Collins et al., 2005; Isozaki et al., 2012].

There are also some pre-ordering methods that do not require a parser. There are BTG-based methods [DeNero and Uszkoreit, 2011; Neubig et al., 2012] and pairwise score-based methods [Tromble and Eisner, 2009; Visweswariah et al., 2011].

2.5 Relationships among Reordering Methods

The joint-ordering methods explained in Section 2.2 are fundamental methods because they are used by the post-ordering and pre-ordering methods.

The distortion models explained in Section 2.2.1 are essential components of phrase-based SMT, which is a joint-ordering method. Distortion models are always used in phrase-based SMT when reordering is necessary.

In the post-ordering framework explained in Section 2.3, phrases are not reordered when translating by phrase-based SMT because translation is first conducted monotonously prior to reordering. Thus, the post-ordering approach does not require a distortion model. However, the post-ordering approach uses a pre-ordering method that reorders a target language sentence into a source language word order to produce training data.

The pre-ordering methods explained in Section 2.4 use distortion models when translating using phrase-based SMT.

2.6 Evaluation

2.6.1 Evaluation Measures

Evaluation of translation quality is challenging. This is because many possible correct translations exist for a given input sentence, and it is difficult to prepare all of the possible correct translations of test sentences in advance. There are two types of evaluation methods for machine translation: automatic evaluation and human evaluation. Since automatic evaluation is not perfect, the reliability of human evaluation is higher than that of automatic evaluation. However, since the cost of human evaluation is high, automatic evaluation is usually used for SMT research.

The most popular automatic evaluation measure for SMT research is BLEU [Papineni et al., 2002]. BLEU is based on n-gram precision, which is the ratio of correct n-grams for each n in relation to the total number of n-grams of the n in the translation results. Here, a correct n-gram means that the n-gram exists in the reference translation. Typically 1 to 4 are used as n of n-grams. This metric is called BLEU-4, which is defined as

$$\text{BLEU-4} = \min\left(1, \frac{\text{output-length}}{\text{reference-length}}\right) \prod_{n=1}^{4} \text{precision}_n,$$

where precision_n represents the precision of n-grams.

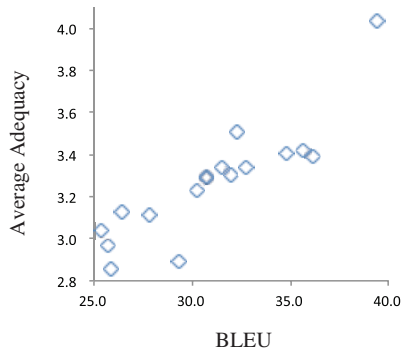

Figure 2.5: Relationship between average adequacy and BLEU for Chinese-English patent translation.

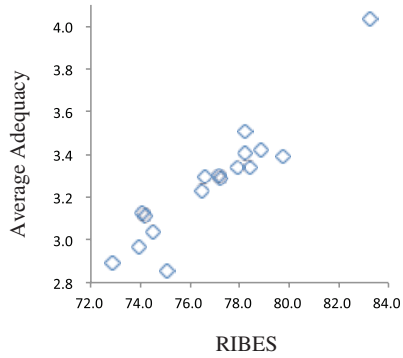

Figure 2.6: Relationship between average adequacy and RIBES for Chinese-English patent translation.

For evaluating translations between languages with largely different word orders, such as Japanese and English, there is an automatic evaluation measure called RIBES [Isozaki et al., 2010a]. RIBES is based on the rank correlation coefficient between the word order of translated outputs and the word order of the reference sentences. For Japanese-English patent translation, RIBES scores were more highly correlated with human evaluation scores than BLEU scores [Isozaki et al., 2010a; Goto et al., 2011; Goto et al., 2013a].

To investigate the relationship between automatic evaluation scores and human evaluation scores, we assessed the evaluation results of the NTCIR-9 Patent

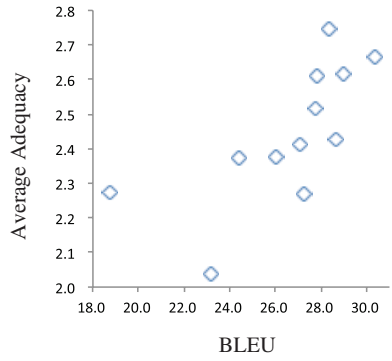

Figure 2.7: Relationship between average adequacy and BLEU for Japanese-English patent translation.

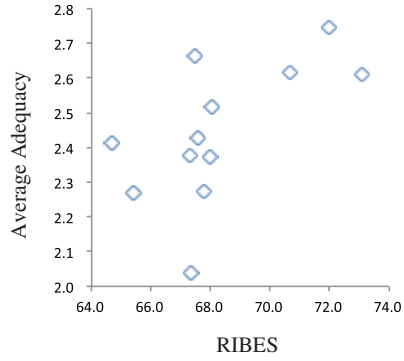

Figure 2.8: Relationship between average adequacy and RIBES for Japanese-English patent translation.

Machine Translation Task [Goto et al., 2011]. In the NTCIR-9 Patent Machine Translation Task, human evaluations based on adequacy were conducted using a 5-point scale (1 to 5) measure. For Chinese-English patent translation, the relationship between the BLEU scores and the average adequacy scores are shown in Figure 2.5 and the relationship between the RIBES scores and the average adequacy scores are shown in Figure 2.6. For Japanese-English patent translation, the relationship between the BLEU scores and the average adequacy scores are shown in Figure 2.7 and the relationship between the RIBES scores and the average adequacy scores are shown in Figure 2.8. We only used scores obtained for

SMT systems. From these figures, there is a correlation between the automatic scores and the average adequacy scores.

2.6.2 Data Sets for Evaluation

Data sets are required to evaluate SMT systems. A data set for SMT evaluations usually consists of the following: parallel sentences and target language sentences for training data, parallel sentences for development data, test sentences, and reference translations of the test sentences. A general evaluation process is as follows. A translation model and a language model are built using the training data, the set of w_i in Equation 2.2 is tuned using the development data, the test sentences are translated by the SMT system, and the system outputs are evaluated using the reference translations.

Some data sets were produced at workshops using shared machine translation tasks. In shared tasks, the efficacy of many machine translation systems of task participants is evaluated using the same data sets and the same evaluation criteria. There are several workshops that have developed data sets for machine translation. For example, there are NIST OpenMT (the domain is mainly news wire and the languages are Arabic-English and Chinese-English.), TIDES and GALE programs (the domain is mainly broadcast news and the languages are Arabic-English and Chinese-English.), IWSLT (the domain is spoken language), WMT (the domains are the proceedings of the European Parliament and news and the languages are European languages.), and NTCIR PATMT/PatentMT (the domain is patent and the languages are Japanese-English, Chinese-English, and English-Japanese.). These data sets facilitate SMT research.

Note that efficacy of each SMT method depends on the source language, the target language, and the domain. Therefore, the efficacy of a method for a specific set of languages and a specific domain does not ensure the efficacy of that method for other languages or domains.

Chapter 3

Distortion Model based on Word Sequence Labeling

3.1 Introduction

Estimating appropriate word order in a target language is one of the most difficult problems for statistical machine translation (SMT). This is particularly true when translating between languages with widely different word orders.

To address this problem, there has been a lot of research done into word reordering: lexical reordering model [Tillman, 2004], which is one of the distortion models, reordering constraints [Zens et al., 2004], pre-ordering [Xia and McCord, 2004], hierarchical phrase-based SMT [Chiang, 2007], and syntax-based SMT [Yamada and Knight, 2001].

In general, source language syntax is useful for handling long distance word reordering. However, obtaining syntax requires a syntactic parser, which is not available for many languages. Phrase-based SMT [Koehn et al., 2007] is a widely used SMT method that does not use a parser.

Phrase-based SMT mainly[1] estimates word reordering using distortion models.[2] Therefore, distortion models are one of the most important components for

[1] A language model also supports estimation.

[2] In this chapter, reordering models for phrase-based SMT, which are intended to estimate the source word position to be translated next in decoding, are called distortion models. This

phrase-based SMT. There are methods other than distortion models for improving word reordering for phrase-based SMT, such as pre-ordering or reordering constraints. However, these methods also use distortion models when translating by phrase-based SMT. Therefore, distortion models do not compete against these methods and are commonly used with them. If a distortion model improves, it will improve the translation quality of phrase-based SMT and will benefit the methods using distortion models.

In decoding by phrase-based SMT, a distortion model estimates the source word position to be translated next (SP) given the last translated source word position (CP). In order to estimate the SP given the CP, many elements need to be considered: the word at the CP, the word at an SP candidate (SPC), the words surrounding the CP and an SPC (context), the relative word order among the SPCs, and the words between the CP and an SPC. In this chapter, these elements are called *rich context*. Even when a parser is unavailable, it is also necessary to consider structural differences between a source language and a target language because the SP depends on structural differences between languages. The major challenge of distortion modeling is consideration of all of the rich context and structural difference between languages.

Previous distortion models could not consider all of the rich context simultaneously. This is because the learning strategy for existing methods was that the models learned probabilities in all of the training data. This meant that the models did not learn preference relations among SPCs in each sentence of the training data. Consequently, it is hard to consider all of the rich context simultaneously using this learning strategy. The MSD lexical reordering model [Tillman, 2004] and a discriminative distortion model [Green et al., 2010] could not simultaneously consider both the word specified at the CP and the word specified at an SPC, or consider relative word order. There is a distortion model that used the word at the CP and the word at an SPC [Al-Onaizan and Papineni, 2006], but, this model did not use context, relative word order, or words between the CP and an SPC. All of these elements are important, and the reasons for their importance

estimation is used to produce a hypothesis in the target language word order sequentially from left to right.

will be detailed in Section 3.2.

In this chapter, we propose a new distortion model consisting of one probabilistic model and which does not require a parser for phrase-based SMT. In contrast to the learning strategy of existing methods, our learning strategy is that the model learns preference relations among SPCs in each sentence of the training data. This leaning strategy enables consideration of all of the rich context simultaneously. Our proposed model, *the sequence model*, can simultaneously consider all of the rich context by identifying the label sequence that specifies the span from the CP to the SP. It enables our model to learn the effect of relative word order among the SPCs as well as learn the effect of distances from the training data. The label sequence can approximately model the structural difference that is considered by synchronous CFG, and the reasons for this will be detailed in Section 3.3.4. Experiments confirmed the effectiveness of our method for Japanese-English, Chinese-English, and German-English translation using NTCIR-9 Patent Machine Translation Task data [Goto et al., 2011], NIST 2008 Open MT task data, and WMT 2008 Europarl data [Callison-Burch et al., 2008].

The rest of this chapter is organized as: Section 3.2 explains the distortion models for phrase-based SMT and previous work, Section 3.3 describes the proposed distortion model, Section 3.4 gives and discusses the experiment results, and Section 3.6 summarizes this chapter.

3.2 Distortion Models for Phrase-Based SMT

A Moses-style phrase-based SMT [Koehn et al., 2007] generates target hypotheses sequentially from left to right. Therefore, the role of the distortion model is to estimate the source phrase position to be translated next whose target side phrase will be located immediately to the right of the already generated hypotheses. An example is shown in Figure 3.1. In Figure 3.1, we assume that only the *kare wa* (English: "he") has been translated. The target word to be generated next will be "bought", and the source word to be selected next will be its corresponding Japanese word *katta*. Thus, a distortion model should estimate phrases including *katta* as a source phrase position to be translated next.

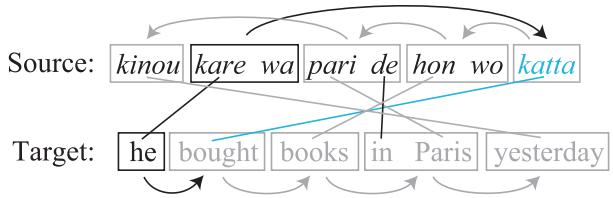

Figure 3.1: An example of left-to-right translation for Japanese-English. Boxes represent phrases and arrows indicate the translation order of the phrases.

To explain the distortion model task in more detail, we need to redefine two terms more precisely, the *current position* (CP) and *subsequent position* (SP) in the source sentence. CP is the source sentence position corresponding to the rightmost aligned target word in the generated target word sequence. SP is the source sentence position corresponding to the leftmost aligned target word in the target phrase to be generated next. The task of the distortion model is to estimate the SP[3] from SP candidates (SPCs) for each CP.[4]

It is difficult to estimate the SP. Figure 3.2 shows examples of sentences that are similar yet have different SPs, with the superscript numbers indicating the word position in the source sentence.

In Figure 3.2(a), the SP is 8. However, in 3.2(b), the word (*kare*) at the CP is the same as 3.2(a), but the SP is different (the SP is 10). From these example sentences, we see that distance is not the essential factor in deciding an SP. We can also see that the word at the CP alone is not enough to estimate the SP. Thus, it is not only the word at the CP, but also the word at an SP candidate (SPC) that should be considered simultaneously.

In Figures 3.2(c) and 3.2(d) , the word (*kare*) at the CP is the same and *karita* (borrowed) and *katta* (bought) are at the SPCs. *Karita* is the word at the SP for 3.2(c), while *katta*, not *karita*, is the word at the SP for 3.2(d). One of the

[3]SP is not always one position, because there may be multiple correct hypotheses.

[4]This definition is slightly different from that of existing methods, such as Moses [Koehn et al., 2007] and Green et al. [2010]. In existing methods, CP is the rightmost position of the last translated source phrase and SP is the leftmost position of the source phrase to be translated next. Note that existing methods do not consider word-level correspondences.

Figure 3.2: Examples of CP and SP for Japanese-English translation. The upper sentence is the source sentence and the sentence underneath is a target hypothesis for each example. The SP is in bold, and the CP is in bold italics. The point of an arrow with an × mark indicates a wrong SP candidate.

reasons for this difference is the relative word order between words. Thus, we can see that considering relative word order, not just looking at what the word at the SP is, is important for estimating the SP.[5]

[5]We checked the probability of a relatively close word position being the SP by using the NTCIR-9 JE data [Goto et al., 2011]. We made lists of words at the SP for each word at the CP in the training data. When a sentence contains two or more words that are included in the list for each word at the CP, and their orientations are the same as that of the SP, we extracted those word pairs from these words. For example, when Figures 3.2(c) and 3.2(d) are the training data, the list of words at the SP for *kare* at the CP consists of *karita* and *katta*. We extract $karita^6$ and $katta^{10}$ as the word pair from Figure 3.2(c), and extract $katta^6$ and $karita^{10}$ as the word pair from Figure 3.2(d). For Figure 3.2(c), the word position relatively close to the CP in

In Figures 3.2(d) and 3.2(e) in, *kare* (he) is at the CP for both, and the word order between *katta* and *karita* are the same. However, the word at the SP for 3.2(d) and the word at the SP for 3.2(e) are different, which shows us that selecting a nearby word is not always correct. The difference is caused by the words surrounding the SPCs (context), the CP context, and the words between the CP and the SPC. Thus, these should all be considered when estimating the SP.

In order to estimate the SP, the following should be considered simultaneously: the word at the CP, the word at an SPC, the relative word order among the SPCs, the words surrounding the CP and an SPC (context), and the words between the CP and an SPC. In other words, rich context should be considered simultaneously.

Returning back to the distribution models, there are distortion models that do not require a parser for phrase-based SMT. The linear distortion cost model used in Moses [Koehn et al., 2007], whose costs are linearly proportional to the reordering distance, always gives a high cost to long distance reordering, even if the reordering is correct. The MSD lexical reordering model [Tillman, 2004; Koehn et al., 2005; Galley and Manning, 2008] only calculates probabilities for the three types of phrase reorderings (monotone, swap, and discontinuous), and does not consider relative word order or words between the CP and an SPC. Thus, these models are not sufficient for long-distance word reordering.

Xiong et al. [2006] proposed distortion models that used context to predict the orientations {left, right} of the SP for their CYK-style decoder. Zens and Ney [2006] proposed distortion models that used context to predict four classes {left, right}×{continuous, discontinuous}. Green et al. [2010] extended the distortion models to use finer classes. [Green et al., 2010]'s model (the outbound model) estimates how far the SP should be from the CP using the word at the CP and its context.[6] Feng et al. [2013] also predicted those finer classes using a CRF model.

the extracted pair is 6 (*karita*[6]). The probability of a word position relatively close to the CP in the extracted pairs being the SP was 81.2%.

[6]They also proposed another model (the inbound model) that estimates reverse direction distance. Each SPC is regarded as an SP, and the inbound model estimates how far the corresponding CP should be from the SP using the word at the SP and its context.

These models do not simultaneously consider both the word specified at the CP and the word specified at an SPC, nor do they consider relative word order.

Al-Onaizan and Papineni [2006] proposed a distortion model that used the word at the CP and the word at an SPC. However, their model did not use context, relative word order, or words between the CP and an SPC.

There is a method that adjusts the linear distortion cost using the word at the CP and its context [Ni et al., 2009]. This model does not simultaneously consider both the word specified at the CP and the word specified at an SPC.

In contrast, our distortion model, the sequence model, addresses the aforementioned issues, utilizes all of the rich context, and approximately considers structural differences between languages.

3.3 Proposed Method

In this section, we first define our distortion model and explain our learning strategy. Then, we describe two models: *the pair model* and *the sequence model*. The pair model is our base model and the sequence model is our main proposed model.

3.3.1 Distortion Model and Learning Strategy

Our distortion model is defined as the model calculating the distortion probability. In this chapter, *distortion probability* is defined as

$$P(X = j|i, S), \tag{3.1}$$

which is the probability of j being the SP, where i is a CP, j is an SPC, S is a source sentence, and X is the random variable of the SP.

We train this model as a discriminative model that discriminates the SP from SPCs. Let J be a set of word positions in S other than i. We train the distortion model subject to

$$\sum_{j \in J} P(X = j|i, S) = 1.$$

The model parameters are learned to maximize the distortion probability of the SP among all of the SPCs J in each source sentence. This learning strategy is a

type of preference relation learning [Evgniou and Pontil, 2002]. In this learning, the distortion probability of the actual SP will be relatively higher than those of all the other SPCs J.

This learning strategy is different from that of Al-Onaizan and Papineni [2006] and Green et al. [2010]. Green et al. [2010], for example, trained their outbound model subject to $\sum_{c \in C} P(Y = c|i, S) = 1$, where C is a set of nine distortion classes[7] and Y is the random variable of the correct distortion class that the correct distortion is classified into. Distortion is defined as $j - i - 1$. Namely, the model probabilities that they learned were the probabilities of distortion classes in all of the training data, not the relative preferences among the SPCs in each source sentence.

3.3.2 Pair Model

The pair model, which is our base model, utilizes the word at the CP, the word at an SPC, and the context of the CP and the SPC simultaneously to estimate the SP. This can be done using our distortion model definition and the learning strategy described in the previous section.

In this work, we use the maximum entropy method [Berger et al., 1996] as a discriminative machine learning method. The reason for this is that a model based on the maximum entropy method can calculate probabilities. However, if we use scores as an approximation of the distortion probabilities, various discriminative machine learning methods can be applied to build the distortion model.

Let s be a source word and $s_1^n = s_1 s_2 ... s_n$ be a source sentence. We add a beginning of sentence (BOS) marker to the head of the source sentence and an end of sentence (EOS) marker to the end, so the source sentence S is expressed as s_0^{n+1} ($s_0 = \text{BOS}$, $s_{n+1} = \text{EOS}$). Our distortion model calculates the distortion probability for an SPC $j \in \{j|1 \leq j \leq n+1 \ \wedge \ j \neq i\}$ for each CP $i \in \{i|0 \leq i \leq n\}$

[7] $(-\infty, -8]$, $[-7, -5]$, $[-4, -3]$, -2, 0, 1, $[2, 3]$, $[4, 6]$, and $[7, \infty)$. In Green et al. [2010], -1 was used as one of distortion classes. However, -1 represents the CP in our definition, and CP is not an SPC. Thus, we shifted all of the distortion classes for negative distortions by -1.

$$P(X = j|i, S) = \frac{1}{Z_i} \exp\left(\mathbf{w}^{\mathrm{T}} \mathbf{f}\left(i, j, S, o, d\right)\right), \tag{3.2}$$

where

$$o = \begin{cases} 0 & (i < j) \\ 1 & (i > j) \end{cases},$$

$$d = \begin{cases} 0 & (|j - i| = 1) \\ 1 & (2 \leq |j - i| \leq 5) \\ 2 & (6 \leq |j - i|) \end{cases},$$

$$Z_i = \sum_{j \in \{j | 1 \leq j \leq n+1 \ \wedge \ j \neq i\}} \exp\left(\mathbf{w}^{\mathrm{T}} \mathbf{f}\left(i, j, S, o, d\right)\right),$$

$\mathbf{w}$ is a weight parameter vector, and each element of $\mathbf{f}(\cdot)$ is a binary feature function which returns 1 when its feature is matched and if else, returns 0. Z_i is a normalization factor, o is an orientation of i to j, and d is a distance class.

Table 3.1 shows the feature templates used to produce features. A feature is defined as an instance of a feature template. Using example (a) from Figure 3.2 will show some instances of each variable, where $i = 2$ and $j = 8$: $o = 1$, $s_i = \textit{kare}$, $s_{i+1} = \textit{wa}$, $s_j = \textit{katta}$, $t_i = \text{NOUN}$, and $d = 2$. t is the part of speech for s. In this case, a feature of $\langle o, s_i, s_j \rangle$ is $\langle o = 1, s_i = \textit{kare}, s_j = \textit{katta} \rangle$ and a feature of $\langle o, s_{i+1}, s_j \rangle$ is $\langle o = 1, s_{i+1} = \textit{wa}, s_j = \textit{katta} \rangle$.

In Equation (3.2), i, j, and S are used by the feature functions. Thus, Equation (3.2) can utilize features consisting of both s_i, which is the word specified at i, and s_j, which is the word specified at j, or both the context of i and the context of j simultaneously. Distance is considered using the distance class d. Distortion is represented by distance and orientation. The pair model considers distortion using six joint classes of d and o.

3.3.3 Sequence Model

The pair model does not consider relative word order among the SPCs nor all the words between the CP and an SPC. Our main proposed model, *the sequence model*, which is described in this section, considers rich context, including relative

Table 3.1: Feature Templates

Template
$\langle o \rangle$, $\langle o, s_{i+p} \rangle^1$, $\langle o, s_{j+p} \rangle^1$, $\langle o, t_i \rangle$, $\langle o, t_j \rangle$, $\langle o, d \rangle$, $\langle o, s_{i+p}, s_{j+q} \rangle^2$, $\langle o, t_i, t_j \rangle$, $\langle o, t_{i-1}, t_i, t_j \rangle$, $\langle o, t_i, t_{i+1}, t_j \rangle$, $\langle o, t_i, t_{j-1}, t_j \rangle$, $\langle o, t_i, t_j, t_{j+1} \rangle$, $\langle o, s_i, t_i, t_j \rangle$, $\langle o, s_j, t_i, t_j \rangle$

Note: t is the part of speech for s.

[1] $p \in \{p | -2 \leq p \leq 2\}$

[2] $(p, q) \in \{(p, q) | -2 \leq p \leq 2 \ \wedge \ -2 \leq q \leq 2 \ \wedge \ (|p| \leq 1 \ \vee \ |q| \leq 1)\}$

Table 3.2: The "C, I, and S" Label Set

Label	Description
C	The current position (CP).
I	A position between the CP and an SPC.
S	A subsequent position candidate (SPC).

word order among the SPCs and including all the words between the CP and an SPC.

In in Figures 3.2(c) and 3.2(d), *karita* (borrowed) and *katta* (bought) both occur in the source sentences. The pair model considers the effect of distances using only the distance class d. If these positions are in the same distance class, the pair model cannot consider the differences in distances. In this case, these are conflict instances during training and it is difficult to distinguish the SP for translation. However, this problem can be solved if the model can consider the relative word order.

The sequence model considers the relative word order. It does this by discriminating the label sequence corresponding to the SP from the label sequences corresponding to each SPC in each sentence. Since each label sequence corresponds to one SPC, if we can identify the label sequence that corresponds to the SP, then we can obtain the SP. The label sequences specify the spans from the CP to each SPC using three kinds of labels that indicate the type of word positions in the spans. The three kinds of labels, "C, I, and S," are shown in Table 3.2. Figure

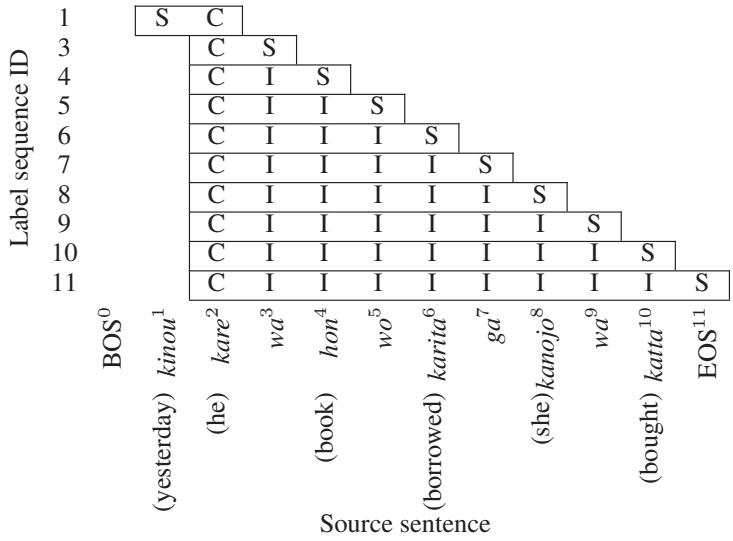

Figure 3.3: Example of label sequences that specify spans from the CP to each SPC for the case of Figure 3.2(c). The labels (C, I, and S) in the boxes are the label sequences.

3.3 shows examples of the label sequences for Figure 3.2(c). The label sequences are represented by boxes and the elements of the sequences are labels. The SPC is used as the label sequence ID for each label sequence.

The label sequence can handle relative word order. Looking at Figure 3.3, the label sequence ID of 10 knows that *karita* exists to the left of the SPC of 10. This is because $karita^6$ carries a label I, while $katta^{10}$ carries a label S, and a position with label I is defined as relatively closer to the CP than a position with label S. By utilizing the label sequence and corresponding words, the model can reflect the effect of *karita* existing between the CP and the SPC of 10 on the probability.

Karita (borrowed) and *katta* (bought) in Figures 3.2(c) and 3.2(d) are not conflict instances in training for the sequence model, whereas they are conflict instances in training for the pair model. The reason is because it is necessary to make the probability of the SPC of 10 smaller than that of the SPC of 6. The pair model tries to make the weight parameters for features with respect to *katta* smaller than those for features with respect to *karita* for 3.2(c), but it also tries to make the weight parameters for features with respect to *karita* smaller than those

for features with respect to *katta* for 3.2(d). Since they have the same features, this causes a conflict. In contrast, the sequence model can give negative weight parameters for the features with respect to the word at the position of 6 with label I, instead of making the weight parameters for the features with respect to the word at the position of 10 with label S smaller than those of 6 with label S.

We use a sequence discrimination technique based on CRF [Lafferty et al., 2001] to identify the label sequence that corresponds to the SP.[8] There are two differences between our task and the CRF task. One difference is that CRF identifies label sequences that consist of labels from all of the label candidates, whereas we constrain the label sequences to sequences where the label at the CP is C, the label at an SPC is S, and the labels between the CP and the SPC are I. The other difference is that CRF is designed for discriminating label sequences corresponding to the same object sequence, whereas we do not assign labels to words outside the spans from the CP to each SPC. However, when we assume that another label such as E has been assigned to the words outside the spans and there are no features involving label E, CRF with our label constraints can be applied to our task. In this chapter, the method designed to discriminate label sequences corresponding to the different word sequence lengths is called *partial CRF*.

The sequence model based on partial CRF is derived by extending the pair model. We introduce the label l and add two extensions to the pair model to identify the label sequences corresponding to the SP. One of the extensions uses labels and the other uses sequence. For the extension using labels, we suppose that label sequences specify the spans from the CP to each SPC using the labels in Table 3.2. We conjoin all the feature templates in Table 3.1 with an additional feature template $\langle l_i, l_j \rangle$ to include the labels into features, where l_i is the label corresponding to the position of i. For example, a feature template of $\langle o, s_{i+1}, s_j, l_i, l_j \rangle$ is derived by conjoining $\langle o, s_{i+1}, s_j \rangle$ in Table 3.1 with $\langle l_i, l_j \rangle$. The

[8]The critical difference between CRFs and maximum entropy Markov models is that a maximum entropy Markov model uses per-state exponential models for the conditional probabilities of next states given the current state, while a CRF has a single exponential model for the joint probability of the entire sequence of labels given the observation sequence [Lafferty et al., 2001].

other extension uses sequence. In the pair model, the position pair of (i, j) is used to derive features. In contrast, to discriminate label sequences in the sequence model, the position pairs of (i, k), $k \in \{k|i < k \leq j \ \vee \ j \leq k < i\}$ and (k, j), $k \in \{k|i \leq k < j \ \vee \ j < k \leq i\}$ are used to derive features. Note that in the feature templates in Table 3.1, i and j are used to specify two positions. When features are used for the sequence model, a value of k is used as one of the two positions. For example, for the position pairs of (i, k), the value of s_k is used as the value of s_j and the value of l_k is used as the value of l_j in the feature template of $\langle o, s_{i+1}, s_j, l_i, l_j \rangle$ to obtain a feature for each k. This is conducted by interpreting the parameters of $\mathbf{f}(\cdot)$ as $\mathbf{f}(i, j, S, o, d, l_i, l_j)$ when the feature templates are used to derive features in the following Equations (3.3) and (3.4).

The distortion probability for an SPC j being the SP given a CP i and a source sentence S is calculated as

$$P(X = j|i, S) = \frac{1}{Z_i} \exp\left(\sum_{k \in M \cup \{j\}} \mathbf{w}^{\mathrm{T}} \mathbf{f}\left(i, k, S, o, d, l_i, l_k\right) + \sum_{k \in M \cup \{i\}} \mathbf{w}^{\mathrm{T}} \mathbf{f}\left(k, j, S, o, d, l_k, l_j\right) \right), \tag{3.3}$$

where

$$M = \begin{cases} \{m|i < m < j\} & (i < j) \\ \{m|j < m < i\} & (i > j) \end{cases}$$

and

$$Z_i = \sum_{j \in \{j|1 \leq j \leq n+1 \ \wedge \ j \neq i\}} \exp\left(\sum_{k \in M \cup \{j\}} \mathbf{w}^{\mathrm{T}} \mathbf{f}\left(i, k, S, o, d, l_i, l_k\right) + \sum_{k \in M \cup \{i\}} \mathbf{w}^{\mathrm{T}} \mathbf{f}\left(k, j, S, o, d, l_k, l_j\right) \right). \tag{3.4}$$

Since j is used as the label sequence ID, discriminating $X = j$ from $X \neq j$ also means discriminating the label sequence ID of the SP from the label sequence IDs of the non-SPs.

The first term in exp(·) in Equation (3.3) considers all of the word pairs located at i and other positions in the sequence, and also their context. The second term in exp(·) in Equation (3.3) considers all of the word pairs located at j and other positions in the sequence, and also their context.

By designing our model to discriminate among different length label sequences, our model can naturally handle the effect of distances. Many features are derived from a long label sequence because it will contain many labels between the CP and the SPC. On the other hand, fewer features are derived from a short label sequence because a short label sequence will contain fewer labels between the CP and the SPC. The bias from these differences provides important clues for learning the effect of distances.[9]

3.3.4 Approximation of Structural Differences by Label Sequences

The label sequence can approximately model structural differences between languages. One method of representing the differences between syntax is synchronous context-free grammar (CFG). Examples of synchronous CFG rules are shown below.[10]

$$\mathrm{VP} \rightarrow \mathrm{NP}_1\ \mathrm{V}_2 \mid \mathrm{V}_2\ \mathrm{NP}_1$$
$$\mathrm{NP} \rightarrow \mathrm{A}_1\ \mathrm{N}_2\ \textit{wo} \mid \mathrm{A}_1\ \mathrm{N}_2$$

The first synchronous CFG rule maps a nonterminal symbol VP to a pair of symbol sequences. The left side of the "|" is associated with the source language (Japanese) and the right side of the "|" is associated with the target language (English). The indexes for the generated nonterminal symbols represent the correspondences of nonterminal symbols between languages. The first rule indicates

[9] Note that the sequence model does not only consider larger context than the pair model, but that it also considers labels. The pair model does not discriminate labels, whereas the sequence model uses label S and label I for the positions except for the CP, depending on each situation. For example, in Figure 3.3, at position 6, label S is used in the label sequence ID of 6, but label I is used in the label sequence IDs of 7 to 11. Namely, even if they are at the same position, the labels in the label sequences are different. The sequence model discriminates the label differences.

[10] Although PP is usually used as a phrase label for phrases including a post position, we use NP as a phrase label for phrases including the post position, *wa*, *ga*, or *wo* in Japanese, to fit Japanese phrase labels to English phrase labels.

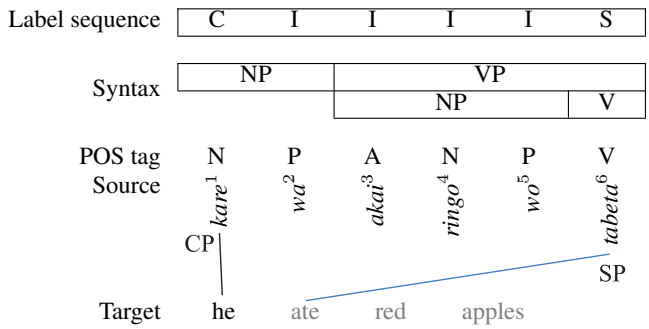

Figure 3.4: The case in which the SP is in a VP.

that when translating a VP, an NP is located to the left side of a V in Japanese, but an NP is located to the right side of a V in English. One of the main characteristics of a VP is that it includes a verb word. Two main characteristics of an NP are as follows: an NP (1) includes a noun word and (2) does not include a verb word when a sentence is a simple sentence. If we can create a model that captures these characteristics, then the model would reflect the differences in syntax between languages.

In a left-to-right decoding process, word reordering is conducted by estimating the SP given the CP in the source sentence. The proposed method models differences in syntax using label sequences that specify the spans from the CP to each SP candidate. We discuss modeling differences in syntax for two cases: the SP is in a VP and the SP is in an NP.

We will first discuss the case in which the SP is in a VP using the example shown in Figure 3.4. In the case of the first synchronous CFG sample rule, the order of generated nonterminal symbols from a VP is reversed order between the languages. Then, the SP is decided at the position of the V (*tabeta*), among the positions in the VP, when the CP is the position *kare*. In the case of a label sequence in Figure 3.4, (1) since the word sequence that corresponds to the label sequence consisting of labels I and S includes a verb word, this part of the label sequence can capture the characteristic of a VP, and (2) since the word sequence that corresponds to the label sequence consisting of labels I includes a noun word but does not include a verb word, this part of the label sequence can capture the

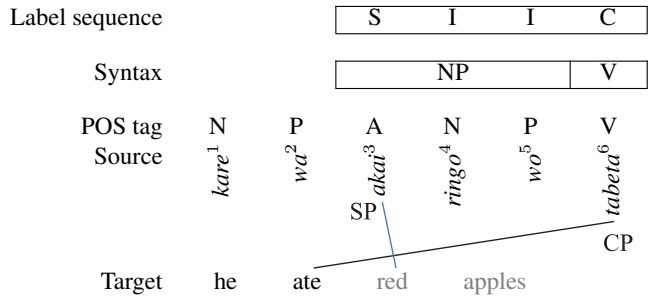

Figure 3.5: The case in which the SP is in an NP.

characteristics of an NP.

We will now discuss the case that the SP is in an NP using an example shown in Figure 3.5. In the case of the second synchronous CFG sample rule, the order of generated nonterminal symbols A and N from an NP is the same order between the languages. Then, the SP is decided at the position of the A (*akai*), among the positions in the NP, when the CP is the position *tabeta*. In the case of a label sequence in Figure 3.5, since the word sequence that corresponds to the label sequence consisting of labels I and S includes a noun word but does not include a verb word, this part of the label sequence can capture the characteristics of an NP.

Since the sequence model utilizes the label sequences that can capture these characteristics of syntax, the model can approximate synchronous CFG rules. Structural differences between languages can be decomposed into synchronous CFG rules. Therefore, the sequence model can approximately model structural differences between languages.

3.3.5 Training Data for Discriminative Distortion Model

In order to train our discriminative distortion model, supervised training data built from a parallel corpus and word alignments between corresponding source words and target words is necessary. Figure 3.6 shows examples of this training data. We create the training data by selecting the target words aligned to the source words sequentially from left to right (target side arrows), then deciding on the order of the source words in the target word order (source side arrows). The

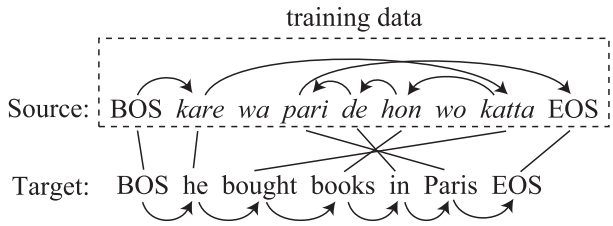

Figure 3.6: Examples of supervised training data. The lines represent word alignments between source words and target words. The English side arrows point to the nearest word aligned on the right.

source sentence and the source side arrows are the training data.

3.4 Experiment

In order to confirm the effects of our distortion model, we conducted a series of Japanese to English (JE), Chinese to English (CE), and German to English (GE) translation experiments.[11]

3.4.1 Data

We used the patent data from the NTCIR-9 Patent Machine Translation Task [Goto et al., 2011] for JE and CE translation. There were 2,000 sentences for the test data and 2,000 sentences for the development data. The reference data is single reference. The translation model was trained using sentences of 40 words or less from the training data. So approximately 2.05 million sentence pairs consisting of approximately 54 million Japanese tokens whose lexicon size was 134k and 50 million English tokens whose lexicon size was 213k were used for JE. Approximately 0.49 million sentence pairs consisting of 14.9 million Chinese tokens whose lexicon size was 169k and 16.3 million English tokens whose lexicon size was 240k were used for CE.

[11] We conducted JE, CE, and GE translation as examples of language pairs with different word orders and of languages where there is a great need for translation into English.

We also used the newswire data from the NIST 2008 Open MT task[12] for CE translation. There were 1,357 sentences for the test data. The reference data is multi-reference (4 references). We used the NIST 2006 test set consisting of 1,664 test sentences as the development data. The translation model was trained using sentences of 40 words or less from the training data. So approximately 2.19 million sentence pairs[13] consisting of 18.4 million Chinese tokens whose lexicon size was 907k and 20.7 million English tokens whose lexicon size was 932k were used.

We used the Europarl data from the WMT 2008 [Callison-Burch et al., 2008] translation task for GE translation. There were 2,000 sentences for the test data. The reference data is single reference. We used the WMT 2007 test set consisting of 2,000 test sentences as the development data. The translation model was trained using sentences of 40 words or less from the training data. So approximately 1.00 million sentence pairs consisting of 20.4 million German tokens whose lexicon size was 226k and 21.4 million English tokens whose lexicon size was 87k were used.

3.4.2 Common Settings

MeCab[14] was used for the Japanese morphological analysis. We adjusted the tokenization of the alphanumeric characters in Japanese to be the same as for the English. The Stanford segmenter[15] and tagger[16] were used for Chinese segmentation and POS tagging and for German POS tagging. GIZA++ and grow-diag-final-and heuristics were used to obtain word alignments. In order to reduce word alignment errors, we removed articles {a, an, the} in English, particles {*ga*, *wo*, *wa*} in Japanese, and articles {der, die, das, des, dem, den, ein, eine, eines, einer, einem, einen} in German before performing word alignments because these func-

[12] To reduce the computational cost, we did not use the comparable corpus (LDC2007T09), the UN corpus (LDC2004E12), or hansard and law domains in the Hong Kong corpus (LDC2004T08).

[13] 1.27 million sentence pairs were from a lexicon (LDC2002L27) and a Named Entity list (LDC2005T34).

[14] `http://mecab.sourceforge.net/`

[15] `http://nlp.stanford.edu/software/segmenter.shtml`

[16] `http://nlp.stanford.edu/software/tagger.shtml`

tion words do not correspond to any words in the other languages (JE and CE) or articles do not always correspond like content words or prepositional words (GE). After word alignment, we restored the removed words and shifted the word alignment positions to the original word positions. We used 5-gram language models with modified Kneser-Ney discounting [Chen and Goodman, 1998] using SRILM [Stolcke et al., 2011]. The language models were trained using the English side of each set of bilingual training data.

We used an in-house standard phrase-based SMT system compatible with the Moses decoder [Koehn et al., 2007]. The phrase table and the lexical distortion model were built using the Moses tool kit. The SMT weighting parameters were tuned by MERT [Och, 2003] using the development data. The tuning was based on the BLEU score [Papineni et al., 2002]. To stabilize the MERT results, we tuned the parameters three times by MERT using the first half of the development data and we selected the SMT weighting parameter set that performed the best on the second half of the development data based on the BLEU scores from the three SMT weighting parameter sets.

We compared systems that used a common SMT feature set from standard SMT features and different distortion model features. The common SMT feature set consists of: four translation model features, phrase penalty, word penalty, and a language model feature. The compared different distortion model features are as follows.

- The linear distortion cost model feature (LINEAR)
- The linear distortion cost model feature and the six MSD bidirectional lexical distortion model [Koehn et al., 2005] features (LINEAR+LEX)
- The outbound and inbound distortion model features discriminating nine distortion classes [Green et al., 2010] (9-CLASS)
- The proposed pair model feature (PAIR)
- The proposed sequence model feature (SEQUENCE)

3.4.3 Training for the Proposed Models

Our distortion model was trained as follows: We used 0.2 million sentence pairs and their word alignments from the data used to build the translation model as the training data for our distortion models. The features that were selected and used were the ones that had been counted[17], using the feature templates in Table 3.1, at least four times for all of the (i, j) position pairs in the training sentences. We conjoined the features with three types of label pairs $\langle l_i = \mathrm{C}, l_j = \mathrm{I}\rangle$, $\langle l_i = \mathrm{I}, l_j = \mathrm{S}\rangle$, or $\langle l_i = \mathrm{C}, l_j = \mathrm{S}\rangle$ to produce features for SEQUENCE. The L-BFGS method [Liu and Nocedal, 1989] was used to estimate the weight parameters of maximum entropy models. The Gaussian prior [Chen and Rosenfeld, 1999] was used for smoothing.[18]

3.4.4 Training for the Compared Models

For 9-CLASS, we used the same training data as for our distortion models. We used the following feature templates to produce features for the outbound model: $\langle s_{i-2}\rangle$, $\langle s_{i-1}\rangle$, $\langle s_i\rangle$, $\langle s_{i+1}\rangle$, $\langle s_{i+2}\rangle$, $\langle t_i\rangle$, $\langle t_{i-1}, t_i\rangle$, $\langle t_i, t_{i+1}\rangle$, and $\langle s_i, t_i\rangle$, where t_i is the part of speech for s_i. These feature templates correspond to the components of the feature templates of our distortion models. In addition to these features, we used a feature consisting of the relative source sentence position as the feature used by Green et al. [2010]. The relative source sentence position is discretized into five bins, one for each quintile of the sentence. For the inbound model[19], i of the feature templates was changed to j. Features occurring four or more times in the training sentences were used. The maximum entropy method with Gaussian prior smoothing was used to estimate the model parameters.

The MSD bidirectional lexical distortion model was built using all of the data used to build the translation model.

[17] When we counted features for selection, we counted features that were from all of the feature templates in Table 3.1 when j was the SP, but we only counted features that were from the feature templates of $\langle s_i, s_j\rangle$, $\langle t_i, t_j\rangle$, $\langle s_i, t_i, t_j\rangle$, and $\langle s_j, t_i, t_j\rangle$ in Table 3.1 when j was not the SP, in order to avoid increasing the number of features.

[18] Let $\mathcal{L}_{\mathbf{w}}$ be the log likelihood of the training data, $\operatorname{argmax}_{\mathbf{w}}(\mathcal{L}_{\mathbf{w}} - \frac{1}{2\sigma^2}\mathbf{w}^T\mathbf{w})$ is used to estimate $\mathbf{w}$. $\sigma^2 = 0.01$ was used for all of the experiments.

[19] The inbound model is explained in footnote 6.

Table 3.3: Japanese-English Translation Evaluation Results for NTCIR-9 Data

	BLEU				RIBES			
Distortion limit	10	20	30	∞	10	20	30	∞
LINEAR	27.98	27.74	27.75	27.30	67.10	67.00	65.89	63.53
LINEAR+LEX	30.25	30.37	30.17	29.98	68.62	68.33	67.31	64.56
9-CLASS	30.74	30.98	30.92	30.75	70.43	69.11	67.97	65.60
PAIR	31.62	32.36	31.96	32.03	70.71	72.04	70.14	68.19
SEQUENCE	32.02	32.96	**33.29**	32.81	71.14	**72.78**	**72.86**	70.55

Table 3.4: Chinese-English Translation Evaluation Results for NTCIR-9 Data

	BLEU				RIBES			
Distortion limit	10	20	30	∞	10	20	30	∞
LINEAR	29.18	28.74	28.31	28.33	75.24	73.46	72.27	71.27
LINEAR+LEX	30.81	30.24	30.16	30.13	75.68	73.54	71.58	70.20
9-CLASS	31.80	31.56	31.31	30.84	77.05	74.43	72.92	71.30
PAIR	32.51	32.30	32.25	32.32	77.75	76.14	74.75	73.93
SEQUENCE	**33.41**	**33.44**	**33.35**	**33.41**	**78.57**	77.67	77.15	76.64

3.4.5 Results and Discussion

We evaluated translation quality based on the case-insensitive automatic evaluation score BLEU-4 [Papineni et al., 2002] and RIBES v1.01 [Isozaki et al., 2010a]. RIBES is an automatic evaluation measure based on word order correlation coefficients between reference sentences and translation outputs. We used distortion limits of 10, 20, 30, and unlimited (∞), which limited the number of words for word reordering to a maximum number for JE and CE. We used distortion limits of 6, 10, and 20 for GE. Our main results are presented in Tables 3.3 to 3.6. The values given are case-insensitive scores. Bold numbers indicate no significant difference from the best result in each language pair and in each evaluation measure using the bootstrap resampling test at a significance level $\alpha = 0.01$ [Koehn, 2004].

The proposed SEQUENCE outperformed the baselines for Japanese to English, Chinese to English, and German to English translation for both BLEU

Table 3.5: Chinese-English Translation Evaluation Results for NIST 2008 Data

	BLEU				RIBES			
Distortion limit	10	20	30	∞	10	20	30	∞
LINEAR	22.50	21.98	21.92	22.09	74.41	71.85	70.92	69.61
LINEAR+LEX	23.29	22.53	23.14	22.85	75.00	72.24	70.67	71.17
9-CLASS	23.30	23.16	22.89	22.98	75.28	73.47	70.26	69.51
PAIR	24.25	23.53	23.87	23.63	**75.88**	73.43	71.20	69.97
SEQUENCE	**24.67**	**24.47**	24.18	**24.34**	**75.92**	73.75	72.42	72.40

Table 3.6: German-English Translation Evaluation Results for WMT 2008 Europarl Data

	BLEU			RIBES		
Distortion limit	6	10	20	6	10	20
LINEAR	26.89	26.59	25.92	78.26	77.83	75.54
LINEAR+LEX	27.09	26.13	26.26	78.38	77.23	75.56
9-CLASS	27.38	27.51	26.97	**78.88**	78.41	76.04
PAIR	**27.87**	27.76	26.89	**78.88**	78.64	75.32
SEQUENCE	**27.88**	**28.04**	27.60	**79.06**	**78.78**	76.74

and RIBES.[20] This demonstrates the effectiveness of the proposed SEQUENCE.[21]

[20]In order to verify the performance of our decoder, we also conducted several experiments for baselines of LINEAR and LINEAR+LEX using the Moses phrase-based decoder. The scores for Moses are follows. LINEAR achived a BLEU score of 27.78 and a RIBES score of 67.08 for JE at distortion limit of 10. LINEAR+LEX achieved a BLEU score of 30.62 and a RIBES score of 69.03 for JE at distortion limit of 20. LINEAR achieved a BLEU score of 22.64 and a RIBES score of 74.73 for CE (NIST 2008) at distortion limit of 10. LINEAR+LEX achieved a BLEU score of 22.85 and a RIBES score of 75.58 for CE (NIST 2008) at distortion limit of 10. These scores and the scores for our decoder were similar.

[21]There are differences in the improvements of the scores from the baselines between the NTCIR-9 results and the NIST 2008 results for CE translation. However, note that when the rates of gains from the baselines are compared, the differences were smaller than the differences of the absolute scores. We think that one of the reasons for the differences is that patent translation is more literal than news translation. If translations are literal, then predicting the subsequent position is easier than with non-literal translations, because there are smaller variations in the

The proposed method is thought to be better than the compared methods for local word ordering since BLEU is sensitive to local word order. The proposed method is also thought to be better than the compared methods for global word ordering since RIBES is sensitive to global word order. The BLEU and RIBES scores of the proposed SEQUENCE were higher than those of the proposed PAIR. This confirms its effectiveness in considering relative word order and words between the CP and an SPC. The proposed PAIR outperformed 9-CLASS for both BLEU and RIBES in most cases[22], confirming that considering both the word specified at the CP and the word specified at the SPC simultaneously was more effective than that of 9-CLASS.

For translating between languages with widely different word orders such as Japanese and English, a small distortion limit is undesirable because there are cases where correct translations cannot be produced with a small distortion limit, since the distortion limit prunes the search space that does not fit within the constraint. Therefore, a large distortion limit is required to translate correctly. For JE translation, our SEQUENCE achieved significantly better results at distortion limits of 20 and 30 than that at a distortion limit of 10 for both BLEU and RIBES, while the baseline systems of LINEAR, LINEAR+LEX, and 9-CLASS did not achieve this. This indicates that SEQUENCE could treat long distance reordering candidates more appropriately than the compared methods.

We also tested hierarchical phrase-based SMT [Chiang, 2007] (HIER) using the Moses implementation [Hoang et al., 2009]. The common data was used to train HIER. We used unlimited max-chart-span for the system setting. Results are given in Table 3.7. Our SEQUENCE outperformed HIER for JE and achieved better than or comparable to HIER for CE and GE. Since phrase-based SMT generally has a faster decoding speed than hierarchical phrase-based SMT, there is merit in achieving better or comparable scores.

translations. This results in a consistency in the subsequent positions in the training data and between the training data and the test set.

[22]There were two cases in which PAIR was worse than 9-CLASS, in Table 3.5 at a distortion limit of 20 and in Table 3.6 at a distortion limit of 20. We think that these were caused by the differences in the SMT weight parameters tuned by MERT.

Table 3.7: Evaluation Results for Hierarchical Phrase-Based SMT

		BLEU	RIBES
HIER	Japanese-English (NTCIR-9)	30.47	70.43
	Chinese-English (NTCIR-9)	32.66	78.25
	Chinese-English (NIST 2008)	23.62	75.86
	German-English (WMT 2008)	27.93	78.78

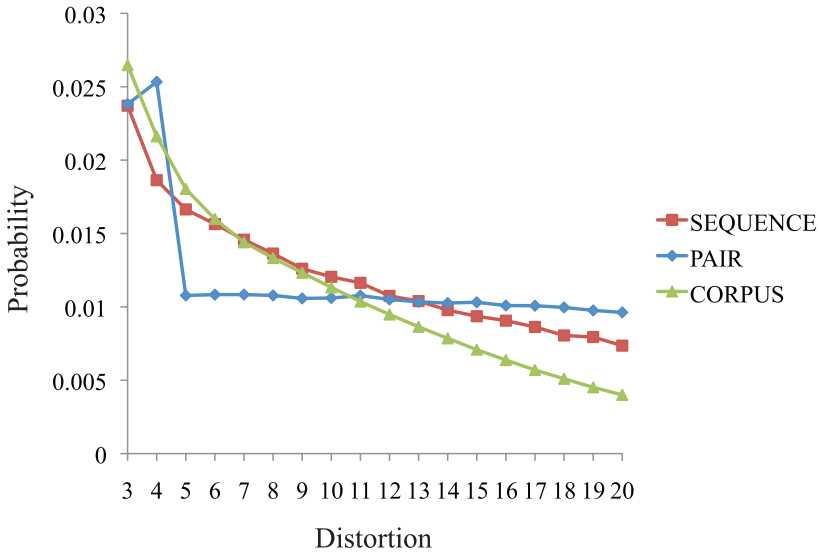

Figure 3.7: Average probabilities for large distortions in Japanese-English translation.

To investigate how well SEQUENCE learns the effect of distance, we checked the average distortion probabilities for large distortions of $j - i - 1$. Figure 3.7 shows three types of probabilities for distortions from 3 to 20 for Japanese-English translation. One type is the average distortion probabilities in the Japanese test sentences for each distortion for SEQUENCE, and another is this for PAIR. The third (CORPUS) is the probabilities for the actual distortions in the training data that were obtained from the word alignments used to build the translation model. The probability for a distortion for CORPUS was calculated by the number of the

distortion divided by the total number of distortions in the training data.

Figure 3.7 shows that when a distance class feature used in the model was the same (e.g., distortions from 5 to 20 had the same distance class feature), PAIR produced average distortion probabilities that were almost the same. In contrast, the average distortion probabilities for SEQUENCE decreased when the lengths of the distortions increased even if the distance class feature was the same, and this behavior was the same as that of CORPUS. This confirms that the proposed SEQUENCE could learn the effect of distances appropriately from the training data.[23]

To investigate the effect of using the words surrounding the SPCs and the CP (context), we conducted experiments without using the words surrounding the SPCs and the CP for PAIR and SEQUENCE. The models without using the surrounding words were trained using only the features that did not contain context. Table 3.8 shows the results for Japanese-English translation.[24] Both the

[23]We also checked the average distortion probabilities for the 9-CLASS outbound model in the Japanese test sentences for Japanese-English translation. We averaged the average probabilities for distortions in a distortion span of [4, 6] and also averaged those in a distortion span of [7, 20], where the distortions in each span are in the same distortion class. The average probability for [4, 6] was 0.058 and that for [7, 20] was 0.165. From CORPUS, the average probabilities in the training data for each distortion in [4, 6] were higher than those for each distortion in [7, 20]. However, the converse was true for the comparison between the two average probabilities for the outbound model. This is because the sum of probabilities for distortions from 7 and above was larger than the sum of probabilities for distortions from 4 to 6 in the training data. This comparison indicates that the 9-CLASS outbound model could not appropriately learn the effects of large distances for JE translation.

[24]Since both the distortion model features with and without the surrounding words represent the same probability shown by Equation (3.1), the same SMT weighting parameters can be used for these features. This was confirmed using SEQUENCE and PAIR, which are also different distortion model features and represent the same probability shown by Equation (3.1). The scores for PAIR with a distortion limit of 30 in Table 3.10 are higher than those in Table 3.3. SEQUENCE was used to tune the SMT weighting parameters in Table 3.10, whereas PAIR was used to tune the SMT weighting parameters in Table 3.3, which indicates that the same SMT weighting parameters can be used for features representing the same probability. However, the SMT weighting parameters tuned by MERT differed for each tuning, and these differences had an effect on the results. For example, the scores for SEQUENCE with a distortion limit of 20 in Tables

Table 3.8: Japanese-English Evaluation Results without and with the Words Surrounding the SPCs and the CP (context)

	BLEU	RIBES
PAIR without surrounding words	30.01	69.02
PAIR (with surrounding words)	32.36	72.04
SEQUENCE without surrounding words	31.72	70.71
SEQUENCE (with surrounding words)	33.29	72.86

Note: The best distortion limit of 20 for PAIR and the best distortion limit of 30 for SEQUENCE in Table 3.3 were used. The "without" results used the same SMT weighting parameters as those of the "with" results to avoid the effects of differences in SMT weighting parameters.

BLEU and RIBES scores for SEQUENCE without using the words surrounding the SPCs and the CP (context) were lower than those for SEQUENCE using the words surrounding SPCs and the CP (context). There was a 1.5 point difference in the BLEU scores for SEQUENCE. This result confirms that using the words surrounding the SPCs and the CP (context) was very effective.

To investigate the effect of using part of speech tags, we conducted experiments without using part of speech tags for PAIR and SEQUENCE. The models without using part of speech tags were trained using only the features that did not contain part of speech tags. The results of this experiment for Japanese-English translation are shown in Table 3.9. Both the BLEU and RIBES scores for SEQUENCE without using part of speech tags were slightly lower than those using part of speech tags. There was a 0.5 point difference in the BLEU scores for SEQUENCE. This result confirms that using part of speech tags was slightly effective for SEQUENCE.

3.3 and 3.10 differ. This difference was caused by the difference in the SMT weighting parameters. It is therefore important to avoid the effects of differences in SMT weighting parameters for comparison.

Table 3.9: Japanese-English Evaluation Results without and with Part of Speech (POS) Tags

	BLEU	RIBES
PAIR without POS	31.41	70.62
PAIR (with POS)	32.36	72.04
SEQUENCE without POS	32.79	72.21
SEQUENCE (with POS)	33.29	72.86

Note: The best distortion limit of 20 for PAIR in Table 3.3 and the best distortion limit of 30 for SEQUENCE were used. The "without" results used the same SMT weighting parameters as those of the "with" results to avoid the effects of differences in SMT weighting parameters.

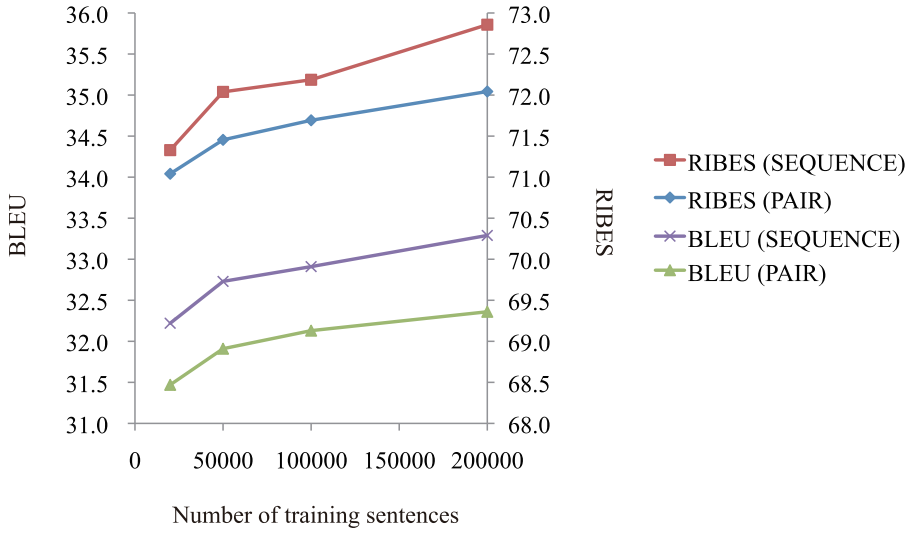

Figure 3.8: Relation between the BLEU/RIBES scores and the number of training sentences of the distortion models for Japanese-English translation.

To investigate the training data sparsity tolerance, we reduced the training data for the sequence model to 100,000, 50,000, and 20,000 sentences for Japanese-English translation.[25] Figure 3.8 show the results for PAIR and SEQUENCE.

[25]We did not conduct experiments using larger training data because there would have been

Table 3.10: Japanese-English Translation Evaluation Results Using the Same SMT Weighting Parameters

	BLEU				RIBES			
Distortion limit	10	20	30	∞	10	20	30	∞
PAIR	31.34	32.29	32.17	32.18	71.12	72.00	70.77	69.16
SEQUENCE	32.24	**33.35**	**33.29**	**33.33**	71.86	**73.75**	72.86	71.60

The best distortion limit of 20 for PAIR and the best distortion limit of 30 for SEQUENCE in Table 3.3 were used. To avoid effects from differences in the SMT weighting parameters, the same SMT weighting parameters used in Table 3.3 were used for each method. SEQUENCE using only 20,000 training sentences achieved a BLEU score of 32.22 and a RIBES score of 71.33. Although the scores are lower than the scores of SEQUENCE with a distortion limit of 30 in Table 3.3, the scores were still higher than those of LINEAR, LINEAR+LEX, and 9-CLASS for JE in Table 3.3. This indicates that the sequence model also works even when the training data is not large. This is because the sequence model considers not only the word at the CP and the word at an SPC but also rich context, and rich context would be effective even on a smaller set of training data.

To investigate the effect of distortion limits for PAIR and SEQUENCE for Japanese-English translation more precisely, we conducted experiments using the same SMT weighting parameters to avoid the effects of differences in SMT weighting parameters. For all of the distortion limits of PAIR and SEQUENCE, we used the same SMT weighting parameters that were used for SEQUENCE with a distortion limit of 30 in Table 3.3, which achieved the best scores in Table 3.3. The results of this are given in Table 3.10.

In Table 3.3, the BLEU score for SEQUENCE with an unlimited distortion was lower than that with a distortion limit of 30. However, Table 3.10 shows that SEQUENCE with an unlimited distortion achieved almost the same BLEU score as that achieved by SEQUENCE with a distortion limit of 30. This indicates that

a very high computational cost to build models using the L-BFGS method.

the difference in BLUE scores for SEQUENCE between a distortion limit of 30 and an unlimited distortion in Table 3.3 was mainly caused by the difference in SMT weighting parameters. However, although the RIBES score for SEQUENCE with an unlimited distortion in Table 3.10 was higher than that in Table 3.3, the RIBES score for SEQUENCE with an unlimited distortion was still lower than that with a distortion limit of 30 in Table 3.10. The RIBES score for SEQUENCE with a distortion limit of 30 was also lower than that with a distortion limit of 20 in Table 3.10. This indicates that SEQUENCE could not sufficiently handle long distance reordering over 20 or 30 words. For such long distance reordering, incorporation with methods that consider sentence-level consistency, such as ITG constraint [Zens et al., 2004], would be useful.

3.5 Related Work

In this section, we will discuss related work other than those discussed in Section 3.2. There is a method that uses SMT sparse features to improve reordering in phrase-based SMT [Cherry, 2013]. However, since the training for this method depends on the SMT weight parameter tuning, the sparse features can only learn from the development data for the SMT weight parameter tuning and cannot utilize a large supply of word aligned training data. Thus, they viewed the sparse features as complementary to existing distortion models. In contrast, our model utilizes a large supply of word aligned training data for training, and it can be built independently of the SMT weight parameter tuning. In addition, SMT sparse features do not calculate the probability of an SPC, whereas our model does. Since [Cherry, 2013]'s sparse features learn from the development data and our model learns from the training data with word alignments, if they are used together, then the SMT system can utilize both the development data and the training data with word alignments to learn reorderings.

There are also reordering models that use a parser: a linguistically annotated ITG [Xiong et al., 2008], a model predicting the orientation of an argument with respect to its verb using a parser [Xiong et al., 2012], and an MSD reordering model using a CCG parser [Mehay and Brew, 2012]. However, none of these

methods consider reordering distances. Structural information such as syntactic structures and predicate-argument structures are useful for reordering, but orientations do not handle distances. A distortion model considering distances of distortions is also useful for methods predicting orientations using a parser when a phrase-based SMT is used, which means that our distortion model does not compete against methods predicting orientations using a parser, but would assist them if used together.

There are word reordering constraint methods that use ITG for phrase-based SMT [Zens et al., 2004; Feng et al., 2010; Cherry et al., 2012]. These methods consider sentence level consistency with respect to ITG. The ITG constraint does not consider distances of reordering and is used with other distortion models. Our distortion model does not consider sentence level consistency, so our distortion model and ITG constraint methods are thought to be complementary.

There are pre-ordering methods using a supervised parser [Xia and McCord, 2004; Wang et al., 2007; Isozaki et al., 2010b; Dyer and Resnik, 2010; Ge, 2010; Genzel, 2010] and methods that do not require a supervised parser [DeNero and Uszkoreit, 2011; Visweswariah et al., 2011; Neubig et al., 2012]. These methods are not distortion models, and a distortion model would be useful for their methods when a phrase-based SMT is used for translation.

There are also tree-based SMT methods [Yamada and Knight, 2001; Chiang, 2007; Galley et al., 2004; Liu et al., 2006; Shen et al., 2008; Huang et al., 2006; Liu et al., 2009; Chiang, 2010]. In many cases, tree-based SMT methods do not use distortion models that consider reordering distance apart from translation rules, because using distortion scores that consider the distances for decoders which do not generate hypotheses from left to right is not trivial. Our distortion model might contribute to tree-based SMT methods if it could be applied to these methods. Investigating the effects will be for future work.

3.6 Summary

In this chapter, we described our distortion models for phrase-based SMT. Our sequence model consists of only one probabilistic model, but it can consider rich

context and can approximately model structural differences between languages without a parser. Unlike the learning strategy used by existing methods, our learning strategy is that the model learns preference relations among SPCs in each sentence of the training data. This learning strategy enables consideration of all of the rich context simultaneously. Experiments indicated that our models achieved better performance than previous models, as measured by both BLEU and RIBES for Japanese-English, Chinese-English, and German-English translation, and also that the sequence model could learn the effect of distances appropriately. Since our models do not require a parser, they can be applied to many languages. Future work includes incorporation into ITG constraint methods and other reordering methods, and application to tree-based SMT methods.

Chapter 4

Post-ordering by Parsing

4.1 Introduction

Reordering target language words into an appropriate word order in the target language is one of the most difficult problems for statistical machine translation (SMT), in particular when translating between languages with widely different word orders such as Japanese and English. In order to handle this problem, a number of reordering methods have been proposed in statistical machine translation research. Those methods can be classified into the following three types.

- *Joint-ordering:* Conducting target word selection and reordering jointly. These methods include phrase-based SMT [Koehn et al., 2003b], hierarchical phrase-based SMT [Chiang, 2007], and syntax-based SMT [Yamada and Knight, 2002; Galley et al., 2004; Quirk et al., 2005; Ding and Palmer, 2005; Liu et al., 2006; Chiang, 2010].

- *Pre-ordering:* First, these methods reorder the source language sentence into a target language word order. Then, they translate the reordered source word sequence using SMT methods [Xia and McCord, 2004; Isozaki et al., 2010b].

- *Post-ordering:* First, these methods translate the source sentence almost monotonously into a target language word sequence. Then, they reorder the target language word sequence into a target language word order [Sudoh et

al., 2011b; Matusov et al., 2005]. In other words, the order of the word reordering and selection processes in post-ordering are the reverse of those in pre-ordering.

Sudoh et al. [2011b] indicated that post-ordering performed better than existing join-ordering methods for Japanese-to-English translations. As for pre-ordering, different translation directions have different reordering problems, even if the language pair is the same, because the performance of pre-ordering methods using a parser depends on the difficulty of estimating the target language word order and the parse accuracy for the source language. In fact, one pre-ordering method for English-to-Japanese translation obtained a large gain, but another pre-ordering method for Japanese-to-English translation could not obtain a large gain [Sudoh et al., 2011a; Goto et al., 2011]. The reason for the high performance of the English-to-Japanese translation is that estimating a Japanese word order based on English is not difficult. This is because Japanese-like word order can be obtained by simply moving an English headword to the end of its syntactic siblings, since Japanese is a typical head-final language [Isozaki et al., 2010b]. On the other hand, English is not a head-final language, which makes estimating English word order more difficult than estimating Japanese word order. Namely, pre-ordering is effective for translating into a target language where estimating word order is not difficult. In contrast, post-ordering is thought to be effective for translating from a source language where estimating word order is not difficult. The reason is as follows: a post-ordering model is built using a parallel corpus consisting of target language sentences and corresponding sentences containing the same words, but in the source language word order. The sentences in the source language word order are produced by changing the target language word order into the source language word order. This change is reliable when estimating source language word order is not difficult.

We employ the post-ordering framework for Japanese-English translation. The post-ordering method consists of a two-step process: (1) almost monotonously translating a Japanese sentence into an English word sequence in a Japanese-like word order; (2) reordering the English word sequence in a Japanese-like word order into an English word order. The first process can be conducted by traditional

phrase-based SMT methods. For the second process, Sudoh et al. [2011b] proposed a method using phrase-based SMT for the English word reordering.

In this chapter, we propose a reordering method based on parsing with inversion transduction grammar (ITG) [Wu, 1997] for the post-ordering framework. The focus of this chapter is the second process of the post-ordering framework, which reorders an English word sequence in a Japanese-like word order into an English word order. Our method uses syntactic structures, which are essential for improving the target word order in translating long sentences between Japanese (a subject-object-verb (SOV) language) and English (an SVO language). Our reordering model parses an English word sequence in a Japanese-like word order using ITG to obtain derivations of Japanese-like syntactic structures, then reorders by transferring the Japanese-like syntactic structures into English structures based on the ITG. Experiments found that our reordering model improved translation quality as measured by both RIBES [Isozaki et al., 2010a] and BLEU [Papineni et al., 2002].

The rest of this chapter is organized as follows. Section 4.2 shows the post-ordering framework and a previous method; Section 4.3 describes the proposed reordering model for post-ordering; Section 4.4 explains the proposed method in detail; Section 4.5 gives and discusses the experiment results; Section 4.6 shows related work; and Section 4.7 concludes.

4.2 Post-ordering for SMT

In this chapter, we take a post-ordering approach [Sudoh et al., 2011b] for Japanese-English translation, which performs translation as a two-step process of word selection and reordering. The translation flow for the post-ordering method is shown in Figure 4.1, where "HFE" is an abbreviation of "Head Final English", which is English words in a Japanese-like structure.[1] The two-step process is as follows.

1. Translating first almost monotonously transforms Japanese into HFE, which

[1]The explanations of pseudo-particles (_va0 and _va2) and other details of HFE is given in Section 4.4.4.

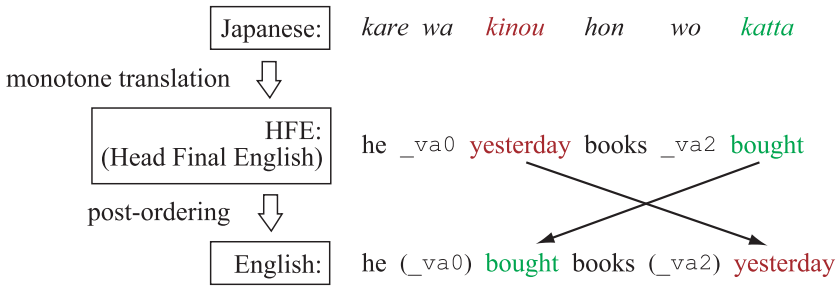

Figure 4.1: Post-ordering framework.

is an English word sequence in almost the same word order as Japanese, using a method such as phrase-based SMT [Koehn et al., 2003b], which can produce accurate translations when only local reordering is required.

2. Reordering then transforms the HFE into English.

In the post-ordering framework, the reordering model that reorders HFE into English is important. Sudoh et al. [2011b] proposed a reordering model that consisted of an HFE-English phrase-based SMT, which reordered by translating an HFE sentence into an English sentence. In general, syntactic structures are important for reordering in translating between languages with widely different word orders. However, the reordering model consisted of phrase-based SMT for post-ordering cannot fully use syntactic structures. In contrast, our reordering model for post-ordering can utilize these useful syntactic structures, which gives our reordering model an advantage.

In order to train a Japanese-HFE SMT model and an HFE-English reordering model, a Japanese-HFE parallel corpus and an HFE-English parallel corpus are needed. These corpora can be constructed by parsing the English sentences in a Japanese-English parallel corpus and applying the head-finalization rules [Isozaki et al., 2010b] to the parsed English sentences. The head-finalization rules change English sentences into HFE sentences, which is in Japanese-like word orders. Then a Japanese-HFE-English parallel corpus is built.

Here, we explain how the head-finalization rules change English into HFE. Japanese is a typical head-final language, where a syntactic head word comes

after nonhead (dependent) words. The head-finalization rules move each syntactic head to the end of its siblings. English sentences are parsed by a parser, Enju [Miyao and Tsujii, 2008], which outputs syntactic heads. Consequently, the parsed English sentences can be reordered into Japanese-like word ordered HFE sentences using the head-finalization rules.

Training for the post-ordering method is conducted via the following steps: first, the English sentences in a Japanese-English parallel corpus are converted into HFE sentences using the head-finalization rules. Next, a monotone phrase-based Japanese-HFE SMT model is built using the Japanese-HFE parallel corpus whose HFE sentences were converted from English sentences. Finally, an HFE-to-English word reordering model is built using the HFE-English parallel corpus.

4.3 Post-ordering Model

In this section, we describe our reordering model for post-ordering, which we concentrate on in this chapter. We explain how the reordering model reorders HFE into English and how to train the reordering model.

4.3.1 Reordering by the ITG Parsing Model

The proposed reordering model for post-ordering, which we have called the *ITG parsing model*, is based on two fundamental frameworks: (i) parsing using probabilistic context free grammar (PCFG) and (ii) the inversion transduction grammar (ITG) [Wu, 1997]. We use syntactic categories for the nonterminals of the ITG.[2] ITG between HFE and English is used as the PCFG for parsing. In this chapter, parsing using ITG is called *ITG parsing*.

We assume that there is an underlying HFE binary tree derivation that produces English word order. The reordering process by the ITG parsing model is shown in Figure 4.2. An HFE sentence is parsed using ITG to obtain an HFE binary tree derivation, which is similar to the syntactic tree structure of the input Japanese sentence. Each nonterminal node that has two child nodes is aug-

[2]This ITG is different from a simple bracketing inversion grammar that has no syntactic content usually employed in ITG parsing.

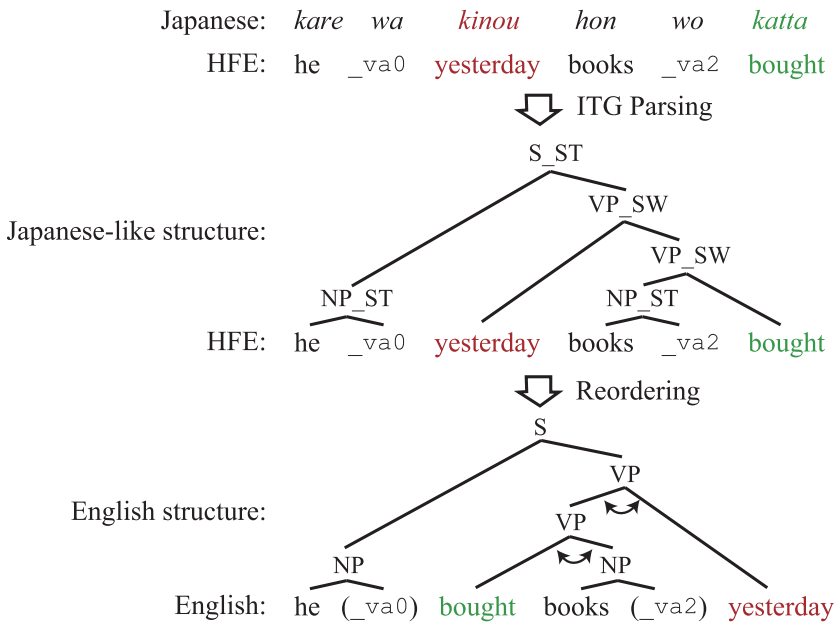

Figure 4.2: Example of post-ordering by parsing.

mented by either an "_ST" (indicating "straight") suffix or an "_SW" (indicating "swap/inversion") suffix. The English word order is determined by the binary tree derivation and the suffixes of the nonterminal nodes. We swap the child nodes of the nodes augmented with the "_SW" suffix in the binary tree derivation in order to produce an English sentence.

4.3.2 Training the ITG parsing model

In order to train the ITG parsing model, the structures of the HFE sentences with "_ST" and "_SW" suffixes are used as the training data. The training data can be obtained from the corresponding English sentences as follows.

First, each English sentence in the training Japanese-English parallel corpus is parsed into a binary tree structure by applying the Enju parser. Then, for each nonterminal node in the English binary tree structure, the two child nodes of each node are swapped if the first child is the head node (see [Isozaki et al., 2010b] for more information on head-finalization rules). At the same time, these nodes with swapped child nodes are annotated with "_SW". When the two child nodes of each

node are not swapped, these nodes are annotated with "_ST". A node with only one child is not annotated with "_ST" or "_SW". The result is an HFE sentence in a binary tree structure augmented with straight or swap/inversion suffixes.

Binary tree structures can be learnable by using an off-the-shelf PCFG learning algorithm. Therefore, HFE binary tree structures can also be learnable. HFE binary tree structures augmented with the straight or swap/inversion suffixes can be regard as derivations of ITG [Wu, 1997] between HFE and English. Therefore, a parsing model learned from the HFE binary tree structures using a PCFG learning algorithm is an ITG model between HFE and English.

In this chapter, we used the state split probabilistic CFG [Petrov et al., 2006] for learning the ITG model. The learned ITG model for parsing is the *ITG parsing model*. The HFE sentences can be parsed by using the ITG parsing model. Then the derivations of the HFE structures can be converted into their corresponding English structures by swapping the child nodes of the nodes with the "_SW" suffix. Note that this ITG parsing model jointly learns how to parse and swap the HFE sentences.

4.4 Detailed Explanation of the translation Method

This section explains the proposed translation method, which is based on the post-ordering framework using the ITG parsing model, in detail.

4.4.1 Derivation of Two-Step Translation

Machine translation is formulated as a problem of finding the most likely target sentence E given a source sentence F.

$$\hat{E} = \underset{E}{\operatorname{argmax}}\, P(E|F).$$

In the post-ordering framework, we divide the translation process into two processes using an HFE sentence M.

$$\begin{aligned}\hat{E} &= \underset{E}{\operatorname{argmax}} \sum_{M} P(E, M|F) \\ &= \underset{E}{\operatorname{argmax}} \sum_{M} P(E|M, F)P(M|F).\end{aligned}$$

The summation is approximated by maximization to reduce computational costs and weighting parameters λ_x (x is r, s, or others) are introduced to be tunable by weighting each model in the same manner as a log-linear model

$$\begin{aligned}\hat{E}, \hat{M} &\approx \underset{E,M}{\operatorname{argmax}} P(E|M, F)P(M|F) \\ &\approx \underset{E,M}{\operatorname{argmax}} P(E|M, F)^{\lambda_r} P(M|F)^{\lambda_s}. \qquad (4.1)\end{aligned}$$

$P(M|F)$ in Equation (4.1) is the probability of translation from a Japanese sentence F into an HFE sentence M. We use the SMT score S of a log-linear SMT model as the logarithm of $P(M|F)^{\lambda_s}$, that is, $\lambda_s \log(P(M|F)) = S$. For the experiment, we used the Moses SMT score [Koehn et al., 2007] from F to M translation as S ($= \lambda_s \log(P(M|F))$). When the Moses SMT score is calculated, feature values, such as a language model probability are scaled by a set of weighting parameters. The set of weighting parameters are usually tuned by a tuning algorithm (e.g., minimum error rate training (MERT) [Och, 2003]). λ_s approximately represents the scaling by the set of weighting parameters.

We compared two reordering models for estimating $P(E|M, F)^{\lambda_r}$ in Equation (4.1).

4.4.2 Translation Using Reordering Model 1

The first reordering model is independent of F given M and we assume that an underlying HFE tree derivation T_M, which is augmented with "_SW" and "_ST", produces an English word order

$$\begin{aligned}
\hat{E}, \hat{T}_M, \hat{M} &\approx \underset{E,T_M,M}{\operatorname{argmax}} P(E, T_M|M)^{\lambda_r} P(M|F)^{\lambda_s} \\
&= \underset{E,T_M,M}{\operatorname{argmax}} P(E|T_M, M)^{\lambda_r} P(T_M|M)^{\lambda_r} P(M|F)^{\lambda_s} \qquad (4.2) \\
&\approx \underset{E,T_M,M}{\operatorname{argmax}} P(E|T_M, M)^{\lambda_{r_1}+\lambda_{r_2}} P(T_M|M)^{\lambda_{r_3}} P(M|F)^{\lambda_s} \\
&\approx \underset{E,T_M,M}{\operatorname{argmax}} P(E)^{\lambda_{r_1}} P(E|T_M, M)^{\lambda_{r_2}} P(T_M|M)^{\lambda_{r_3}} P(M|F)^{\lambda_s}. \qquad (4.3)
\end{aligned}$$

We use the ITG parsing model as $P(T_M|M)$. That is, to obtain high probability T_M, we parse M using the ITG parsing model described in Section 4.3.1. Equation (4.2) is approximated by introducing independent weight parameters λ_{r_1}, λ_{r_2}, and λ_{r_3} instead of λ_r to be tunable by weighting each model in the same manner as a log-linear model; dividing $P(E|T_M, M)^{\lambda_r}$ into two models; and omitting conditions of one of the divided models. E is produced from T_M and M deterministically by swapping the child nodes of the nodes with the "_SW" suffix described in Section 4.3.1. This production process is expressed by $P(E|T_M, M)$. Thus, $P(E|T_M, M)^{\lambda_{r_2}}$ is 1 for E produced from T_M deterministically and is 0 for other E. $P(E)$ is the language model probability of an English sentence E.

Here, we explain why we introduce $P(E)$, which has fewer conditions than $P(E|T_M, M)$. (i) In general, actual models used for calculating probabilities are approximations of equations and not perfect. For example, an n-gram language model appropriately smoothed by a liner combination of an n-gram model and an (n−1)-gram model is usually better than a simple n-gram language model based on the maximum likelihood estimation by relative frequencies. (ii) When the architectures of the two models that calculate the probabilities of the same object are quite different, each model can capture different aspects. Therefore, the n-gram language model of $P(E)$ will remedy the deficiencies of the ITG parsing model of $P(T_M|M)$, which should evaluate generative probability of E because the word order of E is produced from T_M determinately.

4.4.3 Translation Using Reordering Model 2

The first reordering model (reordering model 1) is independent of F. If some noise is included in M when M is produced from F using SMT or if tree derivations of M are more ambiguous than tree structures of F, the tree structure of F will be useful in obtaining a tree derivation of M. This is because F is not a translation result, and a correct tree derivation of M is expected to be similar to a correct tree structure of F, since an HFE sentence is regarded as English words in a Japanese structure.

In this section, we introduce the second reordering model that uses a Japanese syntactic structure. The second reordering model uses the maximum probability Japanese syntactic structure T_F and the maximum probability word alignments A between F and M to obtain an underlying HFE tree derivation T_M, and we also assume that T_M produces the following English word order.

$$
\begin{aligned}
\hat{E}, \hat{T}_M, \hat{M} &\approx \underset{E,T_M,M}{\operatorname{argmax}} P(E, T_M, A, T_F|M, F)^{\lambda_r} P(M|F)^{\lambda_s} \\
&= \underset{E,T_M,M}{\operatorname{argmax}} P(E|T_M, A, T_F, M, F)^{\lambda_r} P(T_M|A, T_F, M, F)^{\lambda_r} \\
&\qquad \times P(A|T_F, M, F)^{\lambda_r} P(T_F|M, F)^{\lambda_r} P(M|F)^{\lambda_s} \qquad (4.4) \\
&= \underset{E,T_M,M}{\operatorname{argmax}} P(E|T_M, M)^{\lambda_r} P(T_M|A, T_F, M)^{\lambda_r} \\
&\qquad \times P(A|M, F)^{\lambda_r} P(T_F|F)^{\lambda_r} P(M|F)^{\lambda_s} \qquad (4.5) \\
&= \underset{E,T_M,M}{\operatorname{argmax}} P(E|T_M, M)^{\lambda_r} P(T_M|A, T_F, M)^{\lambda_r} P(M|F)^{\lambda_s} \qquad (4.6) \\
&\approx \underset{E,T_M,M}{\operatorname{argmax}} P(E|T_M, M)^{\lambda_{r_1}+\lambda_{r_2}} P(T_M|A, T_F, M)^{\lambda_{r_3}} P(M|F)^{\lambda_s} \\
&\approx \underset{E,T_M,M}{\operatorname{argmax}} P(E)^{\lambda_{r_1}} P(E|T_M, M)^{\lambda_{r_2}} P(T_M|A, T_F, M)^{\lambda_{r_3}} P(M|F)^{\lambda_s}. \\
& \qquad (4.7)
\end{aligned}
$$

In Equation (4.4), we assume that E is conditionally independent of A, T_F, and F given T_M and M; that T_M is conditionally independent of F given A, T_F, and M; that A is conditionally independent of T_F given M and F; and that T_F is conditionally independent of M given F. $P(T_F|F)$ in Equation (4.5) is

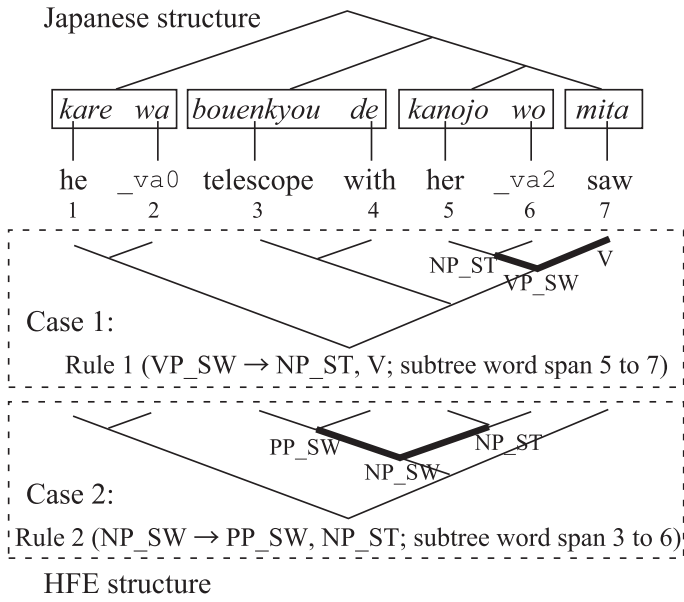

Figure 4.3: Example of subtree spans.

constant given F.[3] $P(A|M,F)$ in Equation (4.5) is approximately assumed as a constant. Equation (4.6) is approximated by introducing independent weight parameters λ_{r_1}, λ_{r_2}, and λ_{r_3} instead of λ_r in the same manner as a log-linear model; dividing $P(E|T_M,M)^{\lambda_r}$ into two models; and omitting conditions of one of the divided models. We use the ITG parsing model with consideration of T_F as $P(T_M|A,T_F,M)$. That is, to obtain high probability T_M, we parse M by the ITG parsing model with consideration of T_F. $P(E|T_M,M)$ represents the deterministic production of E from T_M and M described in Section 4.3.1. $P(E|T_M,M)^{\lambda_{r_2}}$ is 1 for E produced from T_M deterministically and is 0 for other E.

What differs between Equation (4.3) of the previous reordering model 1 and Equation (4.7) of this reordering model 2 is that Equation (4.7) uses $P(T_M|A,T_F,M)$ instead of the $P(T_M|M)$ of Equation (4.3). We use the following simple method using a weighting parameter w $(0 < w < 1)$, which is tuned using development

[3]Note that in these equations, T_F and A are not the argument of the maximum because we use the maximum probability Japanese syntactic structure as T_F and the maximum probability word alignments as A.

data, as one implementation of $P(T_M|A, T_F, M) = P(T_M|A, T_F, M; w)$: a correct T_M is expected to be similar to a correct T_F since an HFE sentence is regarded as English words in a Japanese structure. To reflect this expectation, we change the rule probabilities of the state split PCFG slightly, depending on T_M and T_F using a weighting parameter w $(0 < w < 1)$ as follows.

- If a subtree in T_M does not cross the word span of any subtree in T_F (Rule 1 in Case 1 in Figure 4.3), the rule probability p of the corresponding CFG rule instance is raised to p^w.
- If a subtree in T_M crosses the word span of any subtree in T_F (Rule 2 in Case 2 in Figure 4.3; in this case, the Rule 2 subtree word span 3 to 6 crosses the Japanese subtree word span 5 to 7), the rule probability p of the corresponding CFG rule instance is reduced to p^{2-w}.

p^{2-w} is used to reduce the probability because p^{2-w} is thought to be a symmetric form of p^w, since when w is 1, both p^w and p^{2-w} are the same as p, and as w becomes smaller, the effects increase for both p^w and p^{2-w}. Note that the rule score for each application of the same rule can vary depending on the situation.

Although the resulting rule scores are ad hoc, this assists in making the analysis of T_M closer to T_F.

4.4.4 Head Final English

This section gives more details about Head Final English (HFE) [Sudoh et al., 2011b]. In HFE sentences, the following hold.

1. Each syntactic head is moved toward the end of its siblings except for coordination.
2. Pseudo-particles are inserted after verb arguments: `_va0` (the subject of the sentence head), `_va1` (the subject of a verb), and `_va2` (the object of a verb).
3. Articles (a, an, the) are dropped.

Although these were specified by Sudoh et al. [2011b], we attempt to explain the reasons for the specifications. The reason for (1) is that Japanese is a head-final

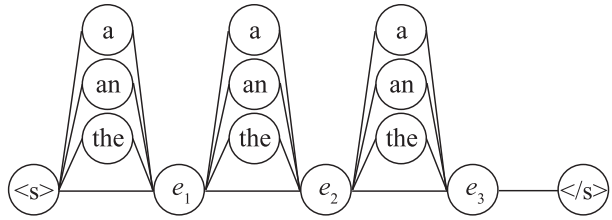

Figure 4.4: Example of a lattice structure.

language. The reasons for (2) and (3) are because translating is usually easier in SMT when words in a parallel sentence correspond one to one than when words correspond one to null. Specifications (2) and (3) try to reduce the one-to-null word correspondences. Japanese sentences contain particles that are case markers for subjects and objects but English has no such corresponding words. The pseudo-particles in HFE correspond to these Japanse particles. On the flip side, Japanese does not contain articles, and thus they are dropped.

There is one point of difference between our HFE construction and that of Sudoh et al. [2011b]: in our method, plural nouns were left as plural instead of being converted to singular, because our reordering model does not change words; it only reorders them.

4.4.5 Article Insertion

Applying our reordering model to an HFE sentence produces an English sentence that does not have articles but does have pseudo-particles. We removed the pseudo-particles from English sentences produced from HFE sentences before calculating the probabilities of $P(E)$ in Equations (4.3) and (4.7) because the language model $P(E)$ without pseudo-particles is simpler than that with pseudo-particles and is more robust than that with pseudo-particles, since E without pseudo-particles is not influenced by insertion errors from inserting pseudo-particles into training data. A language model $P(E)$ was trained from English sentences whose articles were dropped.

In order to output a genuine English sentence E' from E, articles must be inserted into E. A language model trained using genuine English sentences is

used for this purpose. E' is obtained by

$$\hat{E}' = \underset{E' \in S}{\operatorname{argmax}} P(E'),$$

where S is a set consisting of E with articles. We calculate the maximum probability word sequence through a dynamic programing technique for obtaining a genuine English sentence.

Articles are inserted by building a lattice structure which inserts one of the articles {a, an, the} or no article for each word e_i in $E = e_1 e_2 ... e_I$. Figure 4.4 shows the lattice structure in the case of $I = 3$. In Figure 4.4, `<s>` is a special word representing beginning of sentence, and `</s>` is a special word representing end of sentence. The maximum probability word sequence is calculated by applying the Viterbi algorithm for the lattice structure and an n-gram language model.

4.5 Experiment

We investigated the effectiveness of our method by comparing it with other methods for Japanese to English translation.

4.5.1 Setup

We used patent sentence data for the Japanese-to-English translation subtask from the NTCIR-9 [Goto et al., 2011] and NTCIR-8 [Fujii et al., 2010]. The training data and the development data for NTCIR-9 and NTCIR-8 are the same, but the test data is different. There were 2,000 test sentences for NTCIR-9 and 1,251 for NTCIR-8. There were approximately 3.18 million sentence pairs for the training data and 2,000 sentence pairs for the development data. XML entities included in the data were decoded to UTF-8 characters before use.

We used Enju [Miyao and Tsujii, 2008] to parse the English side of the training data. Mecab[4] was used for the Japanese morphological analysis and Cabocha[5] for the Japanese dependency parsing. We adjusted the tokenization of alphanumeric characters and parentheses in Japanese to be the same as for the English. The

[4] http://mecab.sourceforge.net/

[5] http://code.google.com/p/cabocha/

translation model was trained using sentences of 64 words or less from the training data [Sudoh et al., 2011b]. Approximately 2.97 million sentence pairs were 64 words or less. We used 5-gram language models with modified Kneser-Ney discounting [Chen and Goodman, 1998] using SRILM [Stolcke et al., 2011]. The language models were trained using all of the English sentences from the bilingual training data.

We used the Berkeley parser [Petrov et al., 2006], which is an implementation of the state split PCFG based parser, to train the ITG parsing model for HFE and to parse HFE. The ITG parsing model was trained using 0.5 million sentences randomly selected from training sentences of 40 words or less. We performed six split-merge iterations as the same iteration of the parsing model for English [Petrov et al., 2006]. We used the phrase-based SMT system Moses [Koehn et al., 2007] to calculate SMT scores and to produce HFE sentences. The SMT score S was used as the logarithm of $P(M|F)^{\lambda_s}$ in Equation (4.1), that is, $\lambda_s log(P(M|F)) = S$. The distortion limit of the phrase-based SMT was set to 0. With this setting, the phrase-based SMT translates almost monotonously. The SMT weighting parameters were tuned by MERT using the first half of the development data.

For the process of Equation (4.1) through the intermediary M, we used a beam search using the ten-best results of M from Moses outputs. For the processes of parsing M to produce T_M, which is represented by $P(T_M|M)$ in Equation (4.3) and $P(T_M|A, T_F, M, F)$ in Equation (4.7), we used the ten-best parsing results. The probabilities of the ten-best parsing results were approximated to a constant. With this approximation, the value of $P(T_M|M)^{\lambda_{r3}}$ in Equation (4.3) and the value of $P(T_M|A, T_F, M)^{\lambda_{r3}}$ in Equation (4.7) are constant for the ten-best parsing results. Therefore, the value of λ_{r_3} does not affect the results and λ_{r_3} does not need to set for this experiment. As explained in Sections 4.4.2 and 4.4.3, $P(E|T_M, M)^{\lambda_{r2}}$ in Equations (4.3) and (4.7) is 1 for the E produced from T_M deterministically and is 0 for the other E. Therefore, the value of λ_{r_2} does not affect the results and λ_{r_2} does not need to set for this experiment.

Consequently, the parameters to be set for this experiment are λ_{r_1} and w. The parameter λ_{r_1} scales $P(E)$ in Equations (4.3) and (4.7). We used the value of the

weighting parameter for the language model feature in the Japanese-HFE SMT model as the value of λ_{r_1} in order to adjust the scale of $P(E)^{\lambda_{r1}}$ in Equations (4.3) and (4.7) to the scale of $P(M|E)^{\lambda_s}$, which represents the exponent of the score of the Japanese-HFE SMT in Equation (4.1). The parameter w adjusts the strength of the effect from T_F for parsing M for the reordering model 2. w was tuned[6] using the second half of the development data. The tuning was based on the BLEU score [Papineni et al., 2002]. In the experiment, using the Moses SMT score S from F to M translation, we searched for the maximum $\lambda_{r_1} log(P(E)) + S$ in the beam search to obtain $\hat{E}$ for Equations (4.3) and (4.7).

4.5.2 Compared Methods

We used the following 6 comparison methods.

- Phrase-based SMT (PBMT) [Koehn et al., 2003b].
- Hierarchical phrase-based SMT (HPBMT) [Chiang, 2007].
- String-to-tree syntax-based SMT (SBMT) [Hoang et al., 2009].
- Post-ordering based on phrase-based SMT (PO-PBMT) [Sudoh et al., 2011b].
- Post-ordering based on hierarchical phrase-based SMT (PO-HPBMT).
- Post-ordering based on string-to-tree syntax-based SMT (PO-SBMT).

We used Moses [Koehn et al., 2007; Hoang et al., 2009] for these systems. PO-PBMT was the method proposed by Sudoh et al. [2011b]. For PO-PBMT, a distortion limit 0 was used for the Japanese-to-HFE translation, and a distortion limit 20 was used for the HFE-to-English translation. These distortion limit values are the values that achieved the best results in the experiments by Sudoh et al. [2011b]. The PO-HPBMT method changes the post-ordering method of PO-PBMT for the HFE-to-English translation from a phrase-based SMT to a hierarchical phrase-based SMT. The PO-SBMT method changes the post-ordering method of PO-PBMT for the HFE-to-English translation from a phrase-based SMT to a string-to-tree syntax-based SMT. We used a `max-chart-span` of ∞ (unlimited) for

[6] We selected the value of w from $\{0.7, 0.8, 0.9\}$. $w = 0.8$ was used.

the hierarchical phrase-based SMT of PO-HPBMT and the string-to-tree syntax-based SMT of PO-SBMT. We used distortion limits of 12 or 20 for PBMT and `max-chart-span`s of 15 or ∞ (unlimited) for HPBMT and SBMT. For PBMT, a lexicalized reordering model [Koehn et al., 2005], that is, `msd-bidirectional-fe` configuration was used. The default values were used for the other system parameters.

The SMT weighting parameters were tuned by MERT. For PBMT, HPBMT, and SBMT, all of the development data was used for tuning. For the Japanese-to-HFE translation of PO-PBMT, PO-HPBMT, and PO-SBMT, the first half of the development data was used for tuning. For the HFE-to-English translation of PO-PBMT, PO-HPBMT, and PO-SBMT, the following three kinds of data were used for tuning.

- *dev1.* The second half of the development data with HFE produced by translating Japanese using the Japanese-to-HFE SMT.
- *dev1-oracle.* The second half of the development data with HFE that are oracle-HFE made from reference English.
- *dev2-oracle.* The first half of the development data with HFE that are oracle-HFE made from reference English.

4.5.3 Translation Results and Discussion

We evaluated translation quality based on the case-insensitive automatic evaluation scores RIBES v1.01 [Isozaki et al., 2010a] and BLEU-4 [Papineni et al., 2002]. RIBES is an automatic evaluation measure based on the word-order correlation coefficients between reference sentences and translation outputs. The results are shown in Table 4.1.

The method using reordering model 1 described in Section 4.4.2 is "Proposed (without T_F)", and the method using reordering model 2 described in Section 4.4.3 is "Proposed (with T_F)".

We compare the proposed method with T_F to the comparison methods.

First, we made a comparison based on RIBES. For the NTCIR-9 data, the score of the proposed method without T_F was 6.05 points higher than the best

Table 4.1: Evaluation Results

Japanese-to-English	NTCIR-9		NTCIR-8	
	RIBES	BLEU	RIBES	BLEU
Proposed (without T_F)	75.12	32.95	75.91	34.19
Proposed (with T_F)	**75.48**	**33.04**	**76.44**	**34.47**
PBMT (distortion limit 12)	68.61	29.95	68.93	31.01
PBMT (distortion limit 20)	68.28	30.20	69.10	31.26
HPBMT (max chart span 15)	69.98	30.47	70.65	31.32
HPBMT (max chart span ∞)	70.64	30.69	71.65	31.82
SBMT (max chart span 15)	71.28	31.01	71.84	32.00
SBMT (max chart span ∞)	71.84	31.91	72.53	32.73
PO-PBMT (dev1)	67.16	28.75	68.04	30.21
PO-PBMT (dev1-oracle)	69.08	30.01	70.26	31.55
PO-PBMT (dev2-oracle)	68.81	30.39	69.80	31.71
PO-HPBMT (dev1)	70.28	30.54	71.68	32.07
PO-HPBMT (dev1-oracle)	70.54	30.34	71.62	31.89
PO-HPBMT (dev2-oracle)	70.60	30.40	72.13	32.09
PO-SBMT (dev1)	71.80	32.20	73.02	33.21
PO-SBMT (dev1-oracle)	72.52	32.04	73.22	33.21
PO-SBMT (dev2-oracle)	72.31	31.52	72.90	32.76

score from PO-PBMT and 2.60 points higher than the best score from all of the compared methods (the best method was PO-SBMT (dev1-oracle)). For the NTCIR-8 data, it was 5.64 points higher than the best score from PO-PBMT and 2.69 points higher than the best score from all of the compared methods (the best method was PO-SBMT (dev1-oracle)). The proposed method is thought to be better than the compared methods for global word ordering, since RIBES is sensitive to global word order.

Next, we made a comparison based on the widely used BLEU. For the NTCIR-9 data, the score of the proposed method without T_F was 2.56 points higher than

the best score from PO-PBMT and 0.75 points higher than the best score from all of the compared methods (the best method was PO-SBMT (dev1)). For the NTCIR-8 data, it was 2.48 points higher than the best score from PO-PBMT and 0.98 points higher than the best score from all of the compared methods (the best method was PO-SBMT (dev1 and dev1-oracle)). The proposed method is also thought to be better than the compared methods for local word ordering, since BLEU is sensitive to local word order.

The differences between the scores of the proposed method without T_F and the top scores from the compared methods were statistically significant at a significance level of $\alpha = 0.01$ for both RIBES and BLEU, using a bootstrap resampling method [Koehn, 2004] for a statistical significance test. These comparisons demonstrate the effectiveness of the proposed method without T_F for reordering.

When comparing the proposed method with T_F and without T_F, with T_F is higher than without T_F for both RIBES and BLEU for both NTCIR-9 and NTCIR-8. Since the improvements were not large, we calculated a statistical significance test using a bootstrap resampling method [Koehn, 2004] for the differences. For the NTCIR-9 RIBES scores, the difference was statistically significant at a significance level of $\alpha = 0.05$. For the NTCIR-8 RIBES scores, the difference was statistically significant at a significance level of $\alpha = 0.01$. For the NTCIR-9 BLEU scores, the difference was not statistically significant at a significance level of $\alpha = 0.05$, but was statistically significant at a significance level of $\alpha = 0.1$. For the NTCIR-8 BLEU scores, the difference was statistically significant at a significance level of $\alpha = 0.01$. This demonstrates that the method using a Japanese syntactic structure for parsing does have some effectiveness.

In order to investigate the effects of our ITG parsing model more fully, the results with different settings are given here.

We checked different beam widths for the K-best parsing results. Changing the beam widths for K of the K-best parsing results is shown in Figure 4.5 for the NTCIR-9 test data and in Figure 4.6 for the NTCIR-8 test data. The beam width K has a slight effect. However, even when K is 1, that is, only the best parsing results were used, the differences between its RIBES and BLEU scores and the

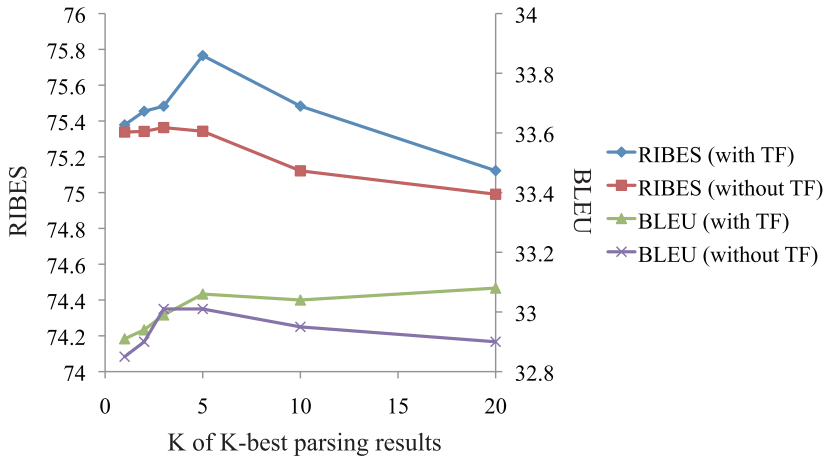

Figure 4.5: Different beam widths K of the K-best parsing results for NTCIR-9.

best scores were not large. This indicates that the top-ranked parsing results were relatively trustworthy compared to the non-top-ranked parsing results. The top ranked parsing results, for example, three- to ten-best, seem almost sufficient.

Figure 4.7 shows the ranking rates of the ten-best parsing results used to produce the final translations for the NTCIR-9 test data[7]. The top-ranked parsing results were used to produce the final translations. This also indicates that the top-ranked parsing results were relatively trustworthy compared to the non-top-ranked parsing results for the following reason: the E of a large $P(E)$ in Equations (4.3) and (4.7) is used to produce the final translation. The English sentence E produced from a correct tree derivation T_M will be a natural English sentence E, whose $P(E)$ will be large, and will be used to produce the final translation.

We checked different beam widths for the N-best results of M. The different beam widths N of the N-best results of M are shown in Figure 4.8 for the NTCIR-9 test data and in Figure 4.9 for the NTCIR-8 test data. From these figures, a beam width of at least 3 is needed to produce the best results, a beam width of 10 is almost sufficient, and a beam width of 50 is thought to be sufficient.

[7]Almost the same results were found for the NTCIR-8 test data, so they are omitted to avoid redundancy.

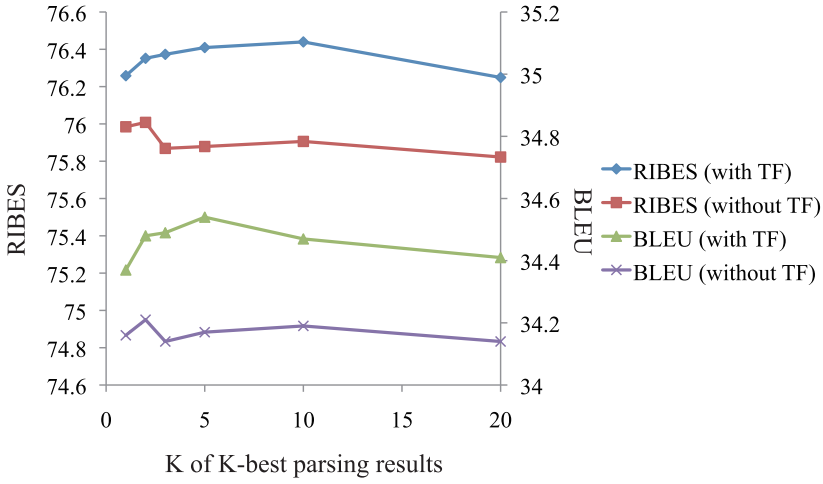

Figure 4.6: Different beam widths K of the K-best parsing results for NTCIR-8.

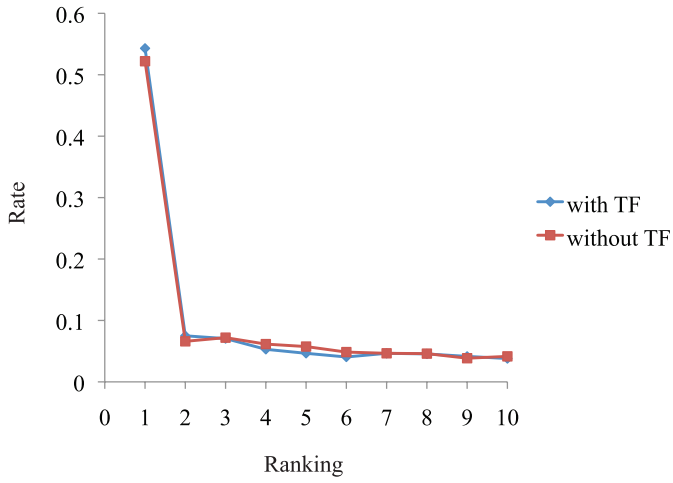

Figure 4.7: The ranking rates of the ten-best parsing results used to produce final translations for NTCIR-9. The vertical axis is the rate of results used to produce final translations and the horizontal axis is the ranking of the ten-best parsing results.

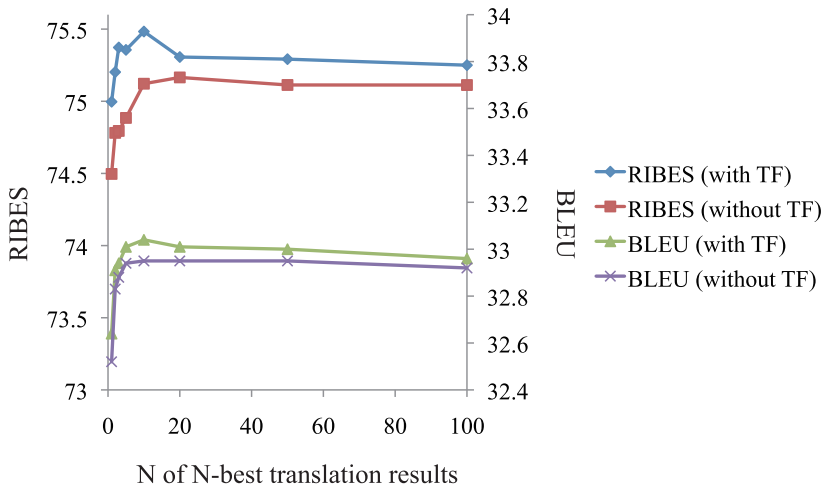

Figure 4.8: Different beam widths N of the N-best translation results for NTCIR-9.

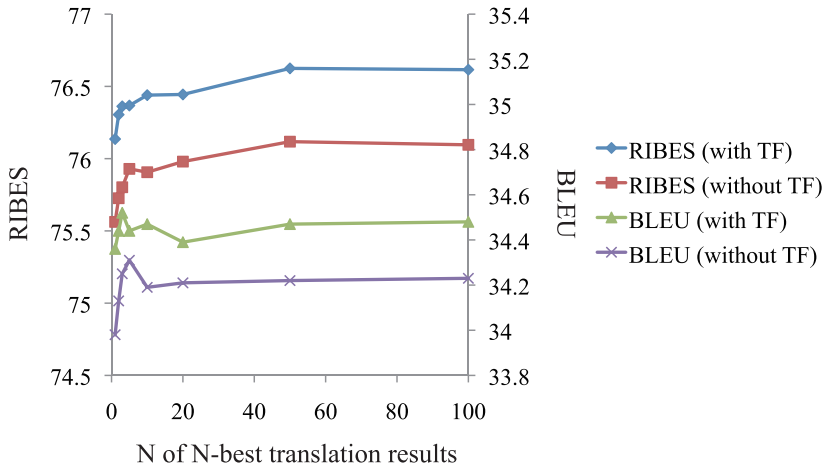

Figure 4.9: Different beam widths N of the N-best translation results for NTCIR-8.

In these experiments, we did not compare our method to pre-ordering methods. However, some groups used pre-ordering methods in the NTCIR-9 Japanese-to-English translation subtask. The NTT-UT group [Sudoh et al., 2011a] used a pre-ordering method that used parsing trees and manually defined pre-ordering rules. The NAIST group [Kondo et al., 2011] used a pre-ordering method [Tromble and Eisner, 2009] that learned a pre-ordering model automatically. These groups were unable to produce both RIBES and BLEU scores that were better than those of the baseline systems of HPBMT and PBMT. In contrast, both the RIBES and BLEU scores for our method were higher than the baseline systems of HPBMT and PBMT. A detailed comparison with pre-ordering methods is our future work.

4.5.4 Results and Discussion Focusing on Reordering

In order to investigate the effects of our post-ordering method more thoroughly, we conducted an "HFE-to-English reordering" experiment which focuses on the effects of word reordering for the post-ordering framework. This experiment confirms the main contribution of our post-ordering method in the framework of post-ordering SMT, as compared with Sudoh et al. [2011b]. In this experiment, we changed the word order of the oracle-HFE sentences made from reference sentences into English using reordering models. This is the same way as in Table 4 in Sudoh et al. [2011b].

Only the test data (input data) differs from the experiment in the previous section. All other settings are the same. In the experiment in Section 4.5.3, Japanese sentences were used for the input data. On the other hand, in the experiment in this section, oracle-HFE sentences were used for the input data. The oracle-HFE sentences were produced by (1) parsing the reference English sentences using the Enju parser and (2) applying the head finalization rules [Isozaki et al., 2010b] to the parsing results. Note that since the oracle-HFE sentences were not produced from Japanese sentences, we only used the proposed method without T_F.

The results are shown in Table 4.2. This results show that our post-ordering method is more effective than PO-PBMT, PO-HPBMT, and PO-SBMT. Since

Table 4.2: Evaluation Results Focusing on Post-Ordering

oracle-HFE-to-English	NTCIR-9		NTCIR-8	
	RIBES	BLEU	RIBES	BLEU
Proposed (without T_F)	**95.33**	**82.58**	**95.59**	**82.78**
PO-PBMT (dev1)	74.89	57.60	75.75	59.03
PO-PBMT (dev1-oracle)	77.79	60.92	78.76	62.33
PO-PBMT (dev2-oracle)	77.34	62.24	78.14	63.14
PO-HPBMT (dev1)	85.26	65.92	86.54	67.40
PO-HPBMT (dev1-oracle)	85.36	66.13	87.07	67.59
PO-HPBMT (dev2-oracle)	84.76	65.28	85.75	66.88
PO-SBMT (dev1)	87.45	69.28	89.98	73.73
PO-SBMT (dev1-oracle)	88.25	69.91	90.99	74.28
PO-SBMT (dev2-oracle)	87.96	68.75	90.52	72.72

RIBES is based on the rank order correlation coefficient, these results show that the proposed method correctly recovered the word order of the English sentences. These high scores also indicate that the parsing results for high quality HFE are fairly trustworthy.

The causes of reordering errors are classified into distinguishing errors between "_ST" and "_SW" and parsing errors. We investigated how often distinguishing errors occurred. We checked the agreement rate of suffixes ("_ST" or "_SW") between the parsing results by the ITG parsing model (parsed trees) and the tree structures of the test data (oracle trees) for the labels with the following conditions: (1) labels that had suffixes ("_ST" or "_SW"); (2) the subtree spans of the labels are the same in the parsed trees and the oracle trees; and (3) labels without suffixes are the same in the parsed trees and the oracle trees. The agreement rate of suffixes was 99.3% for the NTCIR-9 dataset. We checked the number of hidden states learned for the ITG parsing model. The top three labels are VP_ST (61), VP_SW (56), and NP_ST (53). The number in the parenthesis represents the

number of hidden states. Some other major labels are PP_ST (43), S_SW (33), PP_SW (32), S_ST (32), and NP_SW (25). From the high agreement rate, these numbers of hidden states are thought to be enough for learning the distinction between "_ST" and "_SW", and the main cause of errors is thought to be parsing errors. To improve parsing, techniques for parsing such as these of Petrov [2010] will be useful.

Since there are large differences between the values in Table 4.1 and Table 4.2, problems in post-ordering are not entirely solved by improving the reordering accuracy of oracle-HFE. Noise may be included during Japanese-HFE monotone translation. Errors such as word selection errors or lack of translation at the Japanese-HFE monotone translation step cannot be recovered at the reordering step. Using the N-best results for Japanese-HFE monotone translation reduces the effects of these errors compared with using the 1-best result for Japanese-HFE monotone translation. However, this cannot solve the problem perfectly. Word selection is not the only cause of problems. It is rare, but there are word orders in Japanese that cannot be covered by ITG between HFE and English. For example, the fundamental word order of Japanese is SOV, but a word order of OSV is also acceptable in Japanese. An HFE sentence in an OSV word order monotonously translated from a Japanese sentence in an OSV word order cannot be transferred into (S (V O)) by ITG because O and V are not continuous. In this case, it is necessary to convert a Japanese sentence in an OSV word order into a Japanese sentence in an SOV word order at preprocessing.

4.6 Related Work

This section describes related research other than the aforementioned post-ordering [Sudoh et al., 2011b; Matusov et al., 2005]. Features of our method are as follows.

- Monotonously translated sentences are parsed for reordering in the post-ordering framework.
- Word reordering is done by syntactic transfer based on an ITG model merged

with a parsing model.

The post-ordering method splits the word selection and reordering processes. There are many pre-ordering methods that also split the word selection and reordering processes.

Some pre-ordering methods use parsers and manually defined rules for translating different languages. These languages include German to English [Collins et al., 2005], Chinese to English [Wang et al., 2007], English to Hindi [Ramanathan et al., 2008], English to Arabic [Badr et al., 2009], English to Japanese [Isozaki et al., 2010b], and English to five SOV languages (Korean, Japanese, Hindi, Urdu, and Turkish) [Xu et al., 2009]. In English-to-Japanese translation, a pre-ordering method using head finalization rules [Isozaki et al., 2010b], which are used in our post-ordering method, achieved the best quality measured by both RIBES and BLEU, and by the human evaluations which were conducted for the NTCIR-9 patent machine translation task [Sudoh et al., 2011a; Goto et al., 2011]. The reason why this method worked out well is that Japanese is a head-final language, so estimating a Japanese word order based on English is not difficult. On the other hand, English is not a head final language, which makes pre-ordering for Japanese to English more difficult than pre-ordering for the opposite direction, and the pre-ordering method using the head finalization rules cannot be applied. Pre-ordering methods for Japanese to English estimate an English word order based on Japanese. In contrast, the post-ordering methods estimate an English word order based on HFE, which consists of English words. Estimating an English word order based on English words (HFE) is more tractable than estimating an English word order based on Japanese words. This is an advantage of post-ordering methods over pre-ordering methods for Japanese to English translation.

Some pre-ordering methods use parsers and automatically constructed rules [Xia and McCord, 2004; Li et al., 2007; Habash, 2007; Dyer and Resnik, 2010; Ge, 2010; Genzel, 2010; Visweswariah et al., 2010; Wu et al., 2011a; Wu et al., 2011b]. Li et al. [2007] used N-best parsing results. Habash [2007] used labeled dependency structures. Dyer and Resnik [2010] used forests based on parsers. Ge [2010] used a manually-aligned corpus to build a pre-ordering model. Genzel [2010] used a dependency parser and tested English into seven languages, including

Japanese, and German into English. Wu et al. [2011a] investigated the automatic acquisition of Japanese to English pre-ordering rules using bilingual Japanese and English parsing trees. Wu et al. [2011b] used predicate-argument structures to extract pre-ordering rules and tested English to Japanese.

Some pre-ordering methods do not use supervised parsers. Rottmann and Vogel [2007] proposed a pre-ordering method based on POS. Tromble and Eisner [2009] used ITG constraints to reduce computational costs. DeNero and Uszkoreit [2011] and Neubig et al. [2012] proposed methods for inducing binary tree structures automatically from a parallel corpus with high-quality word alignments and using these structures to preorder source sentences based on ITG. They tested English to Japanese, and Neubig et al. [2012] also tested Japanese to English. Visweswariah et al. [2011] trained a model that used pairwise costs of a word by using a small parallel corpus with high-quality word alignments. They tested Hindi to English, Urdu to English, and English to Hindi.

These are all pre-ordering methods, not post-ordering modes, and thus are different from our method.

The post-edit methods also use a two-step translation process that translates first using a rule-based MT system then post-edits the outputs of the rule-based MT using a phrase-based SMT system [Simard et al., 2007; Dugast et al., 2007; Ehara, 2007; Aikawa and Ruopp, 2009], or translates first using a syntax-based SMT system then post-edits the outputs of the syntax-based SMT using a phrase-based SMT system [Aikawa and Ruopp, 2009]. For Japanese-English translation, the first process changes the word order of Japanese into an English word order and translates, then the post-edit process corrects word selection errors from the first process. This method is similar to pre-ordering methods because the first process mainly decides word order and the second process mainly decides word selection. Thus, these post-edit methods are different from our method.

Our method learns the ITG model [Wu, 1997] for reordering. There has also been work done using the ITG model in SMT for joint word selection and reordering. These methods include grammar induction methods from a parallel

corpus [Cherry and Lin, 2007; Zhang et al., 2008; Blunsom et al., 2009; Neubig et al., 2011]; hierarchical phrase-based SMT [Chiang, 2007], which is an extension of ITG; reordering models using ITG [Chen et al., 2009; He et al., 2010]; and ITG constraint for reordering in SMT [Zens et al., 2004; Zhang and Gildea, 2008; Petrov et al., 2008]. Note that the aforementioned methods of DeNero and Uszkoreit [2011] and Neubig et al. [2012] also use ITG for training pre-ordering model. However, none of these methods using the ITG model are post-ordering methods.

Our method uses linguistic syntactic structures for reordering. Linguistic syntactic structures have also been used in various works. There are methods that use target language syntactic structures (string-to-tree) [Yamada and Knight, 2002; Galley et al., 2004; Shen et al., 2008], methods that use source language syntactic structures (tree-to-string) [Quirk et al., 2005; Liu et al., 2006; Huang et al., 2006], and methods that use both the source and the target language syntactic structures (tree-to-tree) [Ding and Palmer, 2005; Liu et al., 2009; Chiang, 2010]. These methods do word selection and reordering simultaneously. In contrast, our method does word selection and reordering separately.

Our method is related to tree-to-tree translation methods using syntactic transfer for word reordering. Since Japanese words and English words do not always correspond one to one, there are large differences between Japanese and English syntactic structures. This makes it difficult to learn syntactic transfer for word reordering. On the other hand, since HFE words and English words always correspond one to one, the difference between HFE and English syntactic structures are smaller than that of Japanese and English. This makes it easier to learn syntactic transfer for word reordering. From these, our method can be regarded to treat a task that learns word reordering based on syntactic transfer for Japanese to English as a more tractable task.

4.7 Summary

This chapter has described a new post-ordering method. Our reordering model consists of a parsing model based on ITG. The proposed method parses sentences that consist of target language words in a source language word order, and does reordering by transferring the syntactic structure similar to the source language syntactic structure into the target language syntactic structure based on ITG. We formulated a method of modeling differences between Japanese syntactic structures and the corresponding English structures that was easily manageable compared with previous methods. It is easier to model differences between HFE syntactic structures and the corresponding English syntactic structures than modeling differences between Japanese syntactic structures and the corresponding English syntactic structures. This is because HFE syntactic structures are perfectly synchronized with the corresponding English syntactic structures, whereas in many cases, some elements of Japanese syntactic structures are not synchronized with the corresponding English syntactic structures. We conducted experiments using Japanese-to-English patent translation. In the experiments, our method outperformed phrase-based SMT, hierarchical phrase-based SMT, string-to-tree syntax-based SMT, and post-ordering methods based on SMT for both RIBES and BLEU. Since RIBES is sensitive to global word order and BLEU is sensitive to local word order, we concluded that the proposed method was better than the compared methods at global word ordering and local word ordering. We also conducted experiments focusing on reordering. These experiments confirmed that our method was able to correctly recover an English word order for high-quality HFE.

Chapter 5

Pre-ordering Using a Target Language Parser

5.1 Introduction

Estimating the appropriate word order for a target language is one of the most difficult problems for statistical machine translation (SMT). This is particularly true when translating between languages with widely different word orders such as Japanese and English. In order to address this problem, there has been a lot of research done on word reordering, such as on: lexicalized reordering model [Tillman, 2004] for phrase-based SMT, hierarchical phrase-based SMT [Chiang, 2007], syntax-based SMT [Yamada and Knight, 2001], pre-ordering [Xia and McCord, 2004], or post-ordering [Sudoh et al., 2011b].

The pre-ordering framework is useful for word reordering because it can utilize source language syntactic structures simply. Specifically, a pre-ordering method using source language syntactic structures for English-to-Japanese translation was confirmed to be highly effective [Sudoh et al., 2011a; Goto et al., 2011]. Existing pre-ordering methods that use source language syntactic structures require a source language syntactic parser. Unfortunately, syntactic parsers are not available for many languages.

As a result of this rack, pre-ordering methods that do not require a parser are useful when there is no source language syntactic parser available [DeNero

and Uszkoreit, 2011; Neubig et al., 2012]. These methods produce pre-ordering rules using word alignments. However, these pre-ordering rules do not utilize syntactic structures, which are one of the essential factors for deciding word order. Therefore, utilizing syntactic structures is the major challenge for pre-ordering methods that do not require a source language syntactic parser.

In this chapter, we propose a novel pre-ordering approach that does not require a source language parser but utilizes syntactic structures using a target language syntactic parser. Source language syntactic structures and corresponding target language syntactic structures are expected to be similar in a parallel corpus [Hwa et al., 2005]. The proposed method utilizes this expectation. We project target language syntactic constituent structures in a parallel corpus to their corresponding source language sentences through word alignments, which produces partial syntactic structures where the words are from the source language but the phrase labels are from the target language syntax. We then construct a probabilistic CFG model and a probabilistic model for unsupervised part-of-speech (POS) tagging using the partial syntactic structures and the Pitman-Yor process. We parse the source language training sentences to produce full binary syntactic tree structures using the produced probabilistic models with the projected partial syntactic structure constraints. A pre-ordering model based on inversion transduction grammar (ITG) [Wu, 1997] is learned using the full binary syntactic constituent structures of the source language sentences and word alignments. Input sentences are parsed using the ITG-based pre-ordering model and their reorderings are also identified jointly.

Our main contributions are (i) a new effective framework for pre-ordering using a target language syntactic parser without requiring a source language syntactic parser, (ii) a simple method for producing full binary syntactic constituent structures of source language sentences from the constituent structures of the corresponding target language sentences using the Pitman-Yor process, and (iii) an empirical confirmation of the effectiveness on Japanese-English and Chinese-English patent translation.

There is a need for translations in situations where a source language parser is not available but a high quality target language parser is available and a source

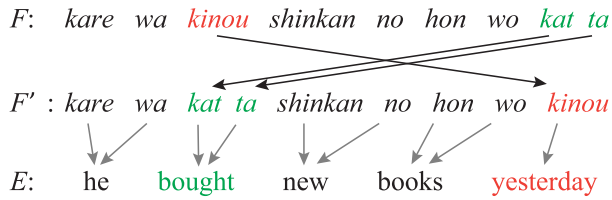

Figure 5.1: Example of pre-ordering for Japanese-English translation.

language word order and a target language word order are largely different such as in subject-object-verb (SOV) and subject-verb-object (SVO) languages. The proposed method can be applied in that situation. In our experiments on Japanese-English and Chinese-English translation using the patent data from the NTCIR-9 and NTCIR-10 Patent Machine Translation Tasks [Goto et al., 2011; Goto et al., 2013a], our method achieved a significant improvement in translation quality as measured by both RIBES [Isozaki et al., 2010a] and BLEU [Papineni et al., 2002] over phrase-based SMT, hierarchical phrase-based SMT, string-to-tree syntax-based SMT, an existing pre-ordering method without requiring a parser, and an existing pre-ordering method using a source language dependency parser.

The rest of this chapter is organized as follows: Section 5.2 shows the pre-ordering framework and previous work, Section 5.3 provides an overview of the proposed method, Section 5.4 explains the training method, Section 5.5 describes the pre-ordering method, Section 5.6 gives and discusses the experiment results, and Section 5.7 concludes the chapter.

5.2 Pre-ordering for SMT

Machine translation is defined as a transformation from a source language sentence F into a target language sentence E. During this process, word reorderings are necessary in many cases. More specifically, long distance word reorderings are necessary when translating between languages with widely different word orders.

For long distance word reorderings, the syntactic structure of F is useful. Pre-ordering is an SMT method that can utilize the syntactic structure of F, which is the approach that we take in this chapter. The pre-ordering approach performs

translation as a two-step process as shown in Figure 5.1. The first process reorders F to F', which is a source language word sequence in almost the same word order as the target language. The second process translates F' into E using an SMT method such as phrase-based SMT, which can produce accurate translations when only local reordering is required.

The pre-ordering framework has been widely studied. Most pre-ordering research reorders word order using reordering rules and the syntactic structure of F obtained by using a source language syntactic parser. Reordering rules are produced automatically [Xia and McCord, 2004; Li et al., 2007; Habash, 2007; Dyer and Resnik, 2010; Ge, 2010; Genzel, 2010; Visweswariah et al., 2010; Wu et al., 2011b; Wu et al., 2011a] or manually [Collins et al., 2005; Wang et al., 2007; Ramanathan et al., 2008; Badr et al., 2009; Xu et al., 2009; Isozaki et al., 2012; Hoshino et al., 2013].

However, if a source language syntactic parser is not available, then these methods cannot be applied. For these cases, pre-ordering methods that do not require a parser would be useful [Tromble and Eisner, 2009; Visweswariah et al., 2011; DeNero and Uszkoreit, 2011; Neubig et al., 2012; Khapra et al., 2013].

Methods inducing a parser deserve particular mention because they are similar to our approach. DeNero and Uszkoreit [2011] and Neubig et al. [2012] induced a non-syntactic parser automatically using a parallel corpus with word alignments. The non-syntactic parser is used to produce binary tree structures of input sentences. The input sentences are then pre-ordered based on the binary tree structures and bracketing transduction grammar (BTG) [Wu, 1997]. The produced binary tree structures are non-syntactic structures. In contrast, our method utilizes syntactic structures for pre-ordering by using a target language syntactic parser.

Compared with non-syntactic structures produced by a non-syntactic parser based on BTG [Neubig et al., 2012], syntactic structures are thought to have advantages in deciding word reorderings for the following two reasons:

- In the syntactic structures, a subtree span is expected to be consistent with the span of an expression whose meanings is cohesive. For example, clauses are thought to be spans whose meanings are cohesive and a clause is ex-

Table 5.1: Comparison of pre-ordering methods based on the necessity of syntactic parsers for source and target languages

Pre-ordering methods	Parser	
	Source	Target
Most of the methods	✓	
[Neubig et al., 2012]		
Proposed method		✓

pressed by a subtree in syntactic structures. In contrast, in the non-syntactic structures produced by BTG, a subtree span is not always consistent with the span of an expression whose meanings is cohesive.

- Syntactic structures have richer information than non-syntactic structures produced by BTG. Syntactic structures have many phrase label types. In contrast, BTG has only one phrase label type.

Therefore, syntactic structures are thought to be useful for performing word reordering for pre-ordering methods.

Table 5.1 compares the necessity of syntactic parsers in the existing and proposed pre-ordering methods for source and target languages. There are cases in which a syntactic parser is not available for the source language but a high quality syntactic parser as available for the target language, and source language word order and target language word order are largely different such as SOV and SVO. Our method is applicable for these cases.

5.3 Overview of the Proposed Method

In this section, we provide an overview of our pre-ordering method.

Our pre-ordering method utilizes syntactic structures using a target language parser even when a source language parser is not available. The syntactic structures of source language sentences and the syntactic structures of the corresponding target language sentences are expected to be similar in a parallel corpus [Hwa

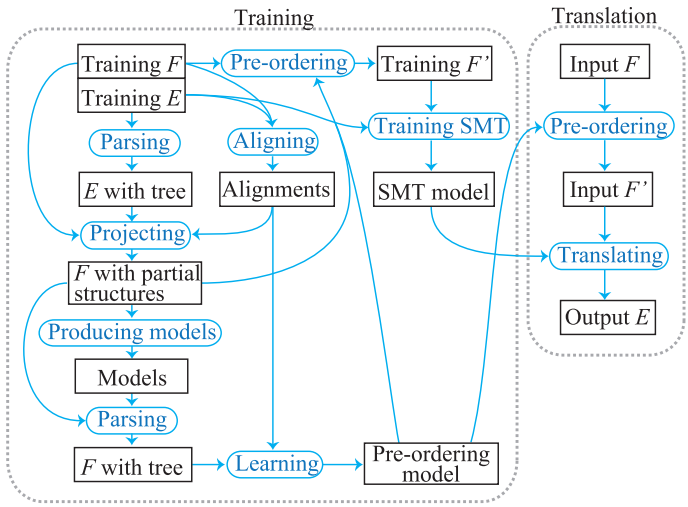

Figure 5.2: The overview of our method.

et al., 2005]. We use this expectation to produce syntactic constituent structures for source language sentences similar to the syntactic constituent structures of target language sentences. Since it is preferable in pre-ordering that the structure for an input sentence be transformable into a structure as similar as possible to the target language structure, the target language syntactic structure of E is suitable for the the syntactic structure of F'. Therefore, it is thought to be suitable for pre-ordering that the syntactic structure of F is based on the target language syntax.

Figure 5.2 shows the overview of our method. Our pre-ordering model is trained as follows:

1. Parsing target language sentences in the training parallel corpus using a syntactic parser to obtain syntactic structures.

2. Projecting the syntactic structures of the target language training sentences to the corresponding source language sentences through word alignments. (Section 5.4.1)

3. Producing a probabilistic CFG model and a probabilistic model for unsupervised POS tagging for the source language using the projected partial syntactic structures. (Section 5.4.2)

4. Parsing the source language training sentences to produce full binary syntactic tree structures using the produced probabilistic models and the projected partial syntactic structures. (Section 5.4.3)

5. Learning the pre-ordering model using the full binary syntactic tree structures and word alignments. (Section 5.4.4)

It is easier to model differences between projected syntactic structures and the corresponding target language syntactic structures than to model differences between non-projected source language syntactic structures and the corresponding target language syntactic structures. This is because the level of synchrony between the projected syntactic structures and the corresponding target language syntactic structures is higher than that between the non-projected source language syntactic structures and the corresponding target language syntactic structures, as the projection produces source language syntactic structures that are maximally synchronized with the corresponding target language syntactic structures.

Input sentences are pre-ordered by jointly parsing and identifying reorderings using the pre-ordering model.

Our main contribution is a new effective framework for pre-ordering using a target language parser. In addition to the main contribution, we propose a new parsing method for a source language without requiring a source language POS tagger or a source language parser.

Jiang et al. [2011] developed a method that projects constituent structures between languages. There are two main differences between our method and theirs. One is the method for estimating the CFG rule probabilities. They count the CFG rules in tree candidates in each sentence for maximum likelihood estimation. In this process, they assume that there is a uniform distribution over the projected tree candidates, and they calculate the expected counts under this assumption. This looks like a single iteration of the EM algorithm. However, their assumption is incorrect. The expected counts of CFG rules in more likely tree candidates should be larger than those of CFG rules in less likely tree candidates. Our method simply solves this problem by using the Pitman-Yor process. The

other difference is in the requirements. Their method requires source language POS tags produced by a POS tagger. In contrast, our method does not require source language POS tags.

Section 5.4 will detail the training method of our pre-ordering model in detail. Section 5.5 will explain the methods for pre-ordering input sentences and the training sentences.

5.4 Training the Pre-ordering Model

In this section, we will explain four components of the training method of our pre-ordering model after parsing target language training sentences.

5.4.1 Projecting Partial Syntactic Structures

We project the binary syntactic constituent structures of the target language sentences in the training parallel corpus onto the corresponding source language sentences through word alignments. Partial syntactic structures of the source language sentences are then obtained. An example of this projection is shown in Figure 5.3.

The projection is conducted by (1) identifying the span in F corresponding to a subtree span in E through word alignments and (2) adding the root phrase label of the subtree in E to the span in F. A span in F is the span from the leftmost position to the rightmost position in the source words that are aligned to a target word in the subtree in E. The root phrase label of a projected subtree in E is added to the projected span in F. Note that if there are non-aligned words adjacent to the span in F, then there is a chance that these words should be contained in the span. That is, when there are non-aligned words adjacent to a span in F, there are ambiguities in the span. A phrase label is added to a span that does not contain the adjacent non-aligned words; that is, phrase labels are added to the spans represented by horizontal solid lines in Figure 5.3.

In this process, in order for the projected structures to compose tree structures and the projected structures to consist of only high quality structures, we do not project any subtree spans in E when their corresponding spans in F conflict with

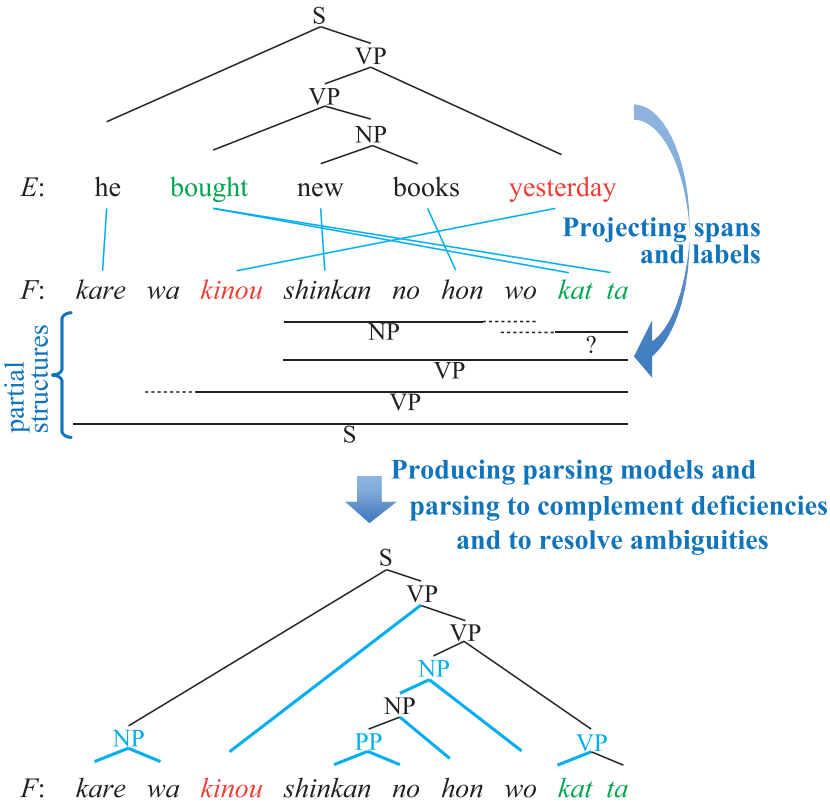

Figure 5.3: Example of projecting syntactic structures from E to F and producing a full binary tree structure. The lines between the words in E and the words in F represent word alignments. The horizontal lines represent projected spans and the labels under the horizontal lines represent their phrase labels. The dotted lines represent ambiguities in the spans. The parts complemented or resolved ambiguities in the structure of F are represented in blue.

each other. Here, the conflict is that two subtree spans that do not overlap in E do overlap, except for non-aligned words, when they are projected to F. That is, we only project the subtree spans of E whose corresponding spans of F are also continuous and do not conflict with each other.

5.4.2 Producing Probabilistic Models for Parsing

The projected structures are usually partial structures. As full binary tree structures are required for learning our pre-ordering model, we produce a probabilistic models for parsing source language sentences in order to produce full binary tree structures.

We will now discuss the method for producing the probabilistic models for parsing in detail. The inputs are a source language sentence F and the projected partial syntactic structures of F described in Section 5.4.1. In this task, the following task characteristics enable use of a simple model to produce full binary tree structures. (i) Partial structures are given. (ii) The set of phrase labels is pre-defined. We also pre-define the number of types of POS tags, which are induced automatically.[1]

For parsing source language structures, we build a probabilistic context free grammar (CFG) model. We use the Pitman-Yor process (PY) [Pitman and Yor, 1997][2] to build the model because its "rich-get-richer" characteristic suits leaning a model from partially annotated structures. We also build a probabilistic model for unsupervised POS tagging using the Pitman-Yor process.

A probabilistic CFG is defined by the 4-tuple $G = (\mathcal{F}, V, S, \mathcal{R})$ where $\mathcal{F}$ is the set of terminals, which are source language words in the training data, V is the set of nonterminals, $S \in V$ is a designated start symbol, and $\mathcal{R}$ is a set of rules. A CFG rule $x \rightarrow \alpha \in \mathcal{R}$ used in this process consists of $x \in V$ and α that consists of two elements in V. V is defined as $V = \mathcal{L} \cup \mathcal{T}$ where $\mathcal{L}$ is the set of phrase labels of the target language syntax, $\mathcal{T} = \{1, 2, ..., |T|\}$ is the set of source language POS tags represented by numbers where $|T|$ is the number of POS tag types, and $\mathcal{L} \cap \mathcal{T} = \varnothing$. Let $f \in \mathcal{F}$ be a source language word and $F = f_1 f_2 ... f_m$. The probability of a derivation tree D is defined as the product of the probabilities

[1] We use numbers as POS tags that are induced automatically. POS tags are also thought to be able to be projected. However, there are some POS tags that cannot be projected. For examples between English and Japanese, determiners exist in English but do not exist in Japanese, and post positions exist in Japanese but not in English. In addition to this, a method without projecting POS tags is simpler than a method that projects POS tags.

[2] Readers unfamiliar with PY can refer to [Teh, 2006] for a detailed description and estimation method for PY.

of its component CFG rules and the probabilities of words given their POS tags as

$$P(D) = \prod_{x \to \alpha \in \mathcal{R}} P(\alpha|x)^{c(x \to \alpha, D)} \prod_{i=1}^{m} P(f_i|t_i), \tag{5.1}$$

where $c(x \to \alpha, D)$ is the number of times $x \to \alpha$ is used for the derivation D, $P(\alpha|x)$ is the probability of generating α given its root phrase label x, $t \in \mathcal{T}$ is a POS tag, index i of t indicates the position in F, and $P(f|t)$ is the probability of generating f given its POS tag t. The designated phrase label, S, is used for the phrase label of the root node of a tree.

Our PY models are distributions over the CFG rules or source language words as follows.

$$P(\alpha|x) \sim \mathrm{PY}_x(d_{\mathrm{cfg}}, \theta_{\mathrm{cfg}}, P_{\mathrm{base}}(\alpha|x)) \text{ and}$$
$$P(f|t) \sim \mathrm{PY}_t(d_{\mathrm{tag}}, \theta_{\mathrm{tag}}, P_{\mathrm{base}}(f|t)),$$

where d_{cfg}, θ_{cfg}, d_{tag}, and θ_{tag} are hyperparameters for the PY models. The hyperparameters are optimized with the auxiliary variable technique [Teh, 2006].[3] The backoff probability distoributions, $P_{\mathrm{base}}(\alpha|x)$ and $P_{\mathrm{base}}(f|t)$, are uniform as follows.

$$P_{\mathrm{base}}(\alpha|x) = \frac{1}{|V|^2} \text{ and}$$
$$P_{\mathrm{base}}(f|t) = \frac{1}{|\mathcal{F}|},$$

where $|V|$ is the number of nonterminal types and $|\mathcal{F}|$ is the lexicon size of source language words in the training data. Since our CFG rule has two leaf nodes, the number of pair nonterminal node types is $|V|^2$.

Sampling for building the distributions is according to Equation (5.1) with the following constraints. When there are projected spans, we constrain the sampling to sample the derivation trees that do not conflict with the projected spans. Here, the conflict is that both a subtree span in the tree derivation and a projected span partially overlap each other. When there are ambiguities in the projected spans, the laxest constraints are applied for each tree derivation. When there is the

[3] We put a prior of Beta$(1, 1)$ on d_{cfg} and d_{tag} and a prior of Gamma$(1, 1)$ on θ_{cfg} and θ_{tag}.

projected phrase label for a subtree span in a derivation tree, we constrain the sampling to sample the projected phrase label.

We use the sentence-level blocked Gibbs sampler. The sampler consists of the following two steps: for each sentence, (1) calculate the inside probability from the bottom up using the inside algorithm, (2) sample a tree from the top down. In the first step, when we calculate the inside probabilities for each phrase label in each cell in the triangular table of the inside algorithm, we save inside probabilities for each CFG rule. In the second step, we sample a CFG rule according to the inside probabilities for the CFG rules in each cell from the top down. In order to reduce computational costs, we only use N-best POS tags for each word when the inside probabilities are calculated. In our experiment in Section 5.6, we used five-best POS tags for each word.

5.4.3 Parsing to Produce Full Binary Tree Structures

After the distributions of the PY models are built, we parse the source language sentences to complement deficiencies and resolve ambiguities in the projected partial structures. We calculate the most likelihood full binary tree structures based on the CYK algorithm within the constraints of the projected spans and phrase labels using the produced probabilistic CFG model and the produced probabilistic model for unsupervised POS tagging. The probability for a derivation tree is calculated using Equation (5.1). The constraints are the same constraints used for the sampling to build the probabilistic models. The produced full binary tree structures consist of the phrase labels of the target language syntax. An example of producing a full binary tree structure is shown in Figure 5.3.

Note that when the full binary trees are produced, it does not mean that all of the projected spans are included in the full binary trees. If there are non-aligned words adjacent to the projected spans, then there may be cases in which the projected spans are not included in the full binary tree. For example, when a projected span is $(f_1 f_2) f_3$ and f_3 is a non-aligned word where parentheses denote a span, a full binary tree may be $(f_1 (f_2 f_3))$, which does not include the projected span, because there are ambiguities in the span when a non-aligned word

is adjacent to the span as explained in Section 5.4.1.

5.4.4 Learning the Pre-ordering Model

We learn our pre-ordering model using the full binary tree structures of source language sentences and word alignments.

The pre-ordering model is a model based on two fundamental frameworks [Goto et al., 2013b]: (i) parsing using probabilistic CFG and (ii) the inversion transduction grammar (ITG) [Wu, 1997]. In this chapter, the model combining (i) and (ii) is called the *ITG parsing model* and parsing using ITG is called *ITG parsing*. We use the ITG parsing model for pre-ordering while Goto et al. [2013b] used this model for post-ordering.

In order to obtain the training data for the pre-ordering model, we first obtain the reordering that produces the word order of F' most similar to the word order of the corresponding E using their word alignments. The reordering is conducted by swapping child nodes in the binary tree structure of F, so that Kendall τ is maximized between F' and E. Figure 5.4 shows an example of the tree structure of F' calculated from the tree structure of F and word alignments.

The nodes whose child nodes are swapped to transform F into F' are then annotated with an "_SW" suffix (indicating "swap/inversion") and other nodes with two child nodes are annotated with an "_ST" suffix (indicating "straight") in the binary tree for F. Figure 5.5 shows an example of F and its binary tree structure annotated with the _ST and _SW suffixes. The result is that the binary tree syntactic structure of F is augmented with straight or swap/inversion suffixes, which can be regarded as a derivation of ITG between F and F'.

Therefore, an ITG model can be learned from the binary tree structures using a probabilistic CFG learning algorithm. This learned model is the ITG parsing model. In this chapter, we use the state split probabilistic CFG [Petrov et al., 2006] for learning the ITG parsing model. The learned ITG parsing model is our pre-ordering model.

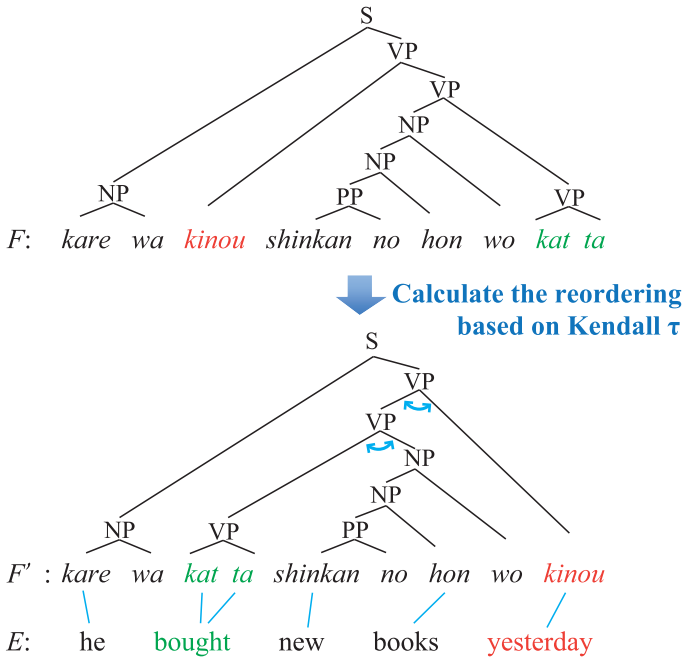

Figure 5.4: Example of calculating the reordering for F' based on Kendall τ.

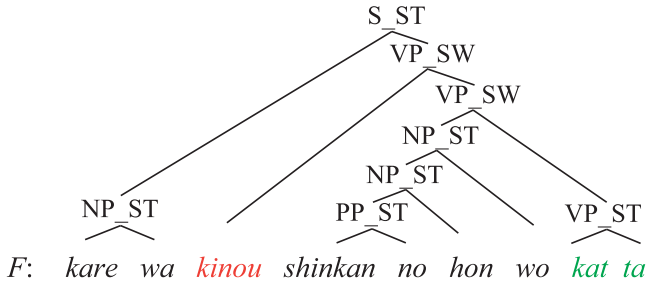

Figure 5.5: Example of F and its binary tree structure annotated with _ST and _SW suffixes.

5.5 Pre-ordering Sentences

This section describes how to pre-order input sentences and the training sentences.

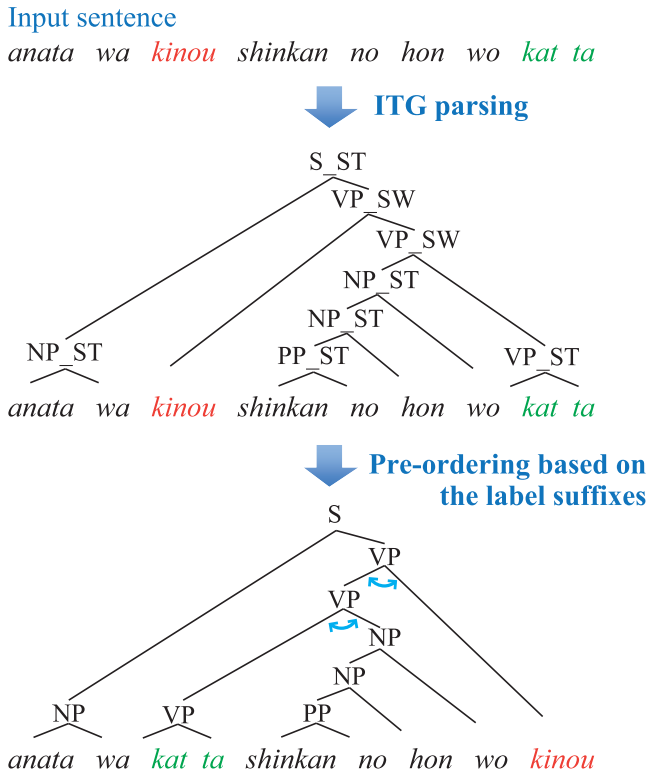

Figure 5.6: Pre-ordering an input sentence.

5.5.1 Pre-ordering Input Sentences

Input sentences are pre-ordered using the ITG parsing model described in Section 5.4.4. The pre-ordering process is shown in Figure 5.6. An input sentence F is parsed using the ITG parsing model. When F is parsed, the reordering for F' is jointly identified based on ITG. Each non-terminal node in the tree derivation is augmented by either an "_ST" suffix or an "_SW" suffix. The word order for F' is determined by the binary tree derivation with the suffixes of the non-terminal nodes. We swap the child nodes of the nodes augmented with the "_SW" suffix in the binary tree derivation in order to produce F'.

5.5.2 Pre-ordering the Training Sentences

After transforming the F of an input sentence into F', phrase-based SMT translates F' into E. Therefore, phrase-based SMT requires parallel F' and E sentences to train its translation model. Now, we will detail how to produce F' in the parallel sentences for training the SMT translation model.

If F' in the training data is produced using the same method as the method for pre-ordering input sentences, then the word order of F' in the training data is consistent with the word order of pre-ordered input sentences. However, the method for pre-ordering input sentences is not always the best method to pre-order the training data. This is because a corresponding E already exists in the training data and we also have to consider the consistency between F' and E in the training data.

There is a reason why we have to consider the consistency between F' and E. The objective of pre-ordering the training sentences is the building of a phrase table, which is the SMT translation model, consisting of parallel phrase pairs between F' and E and their probabilities. When both corresponding expressions in E and F' are continuous, they can be extracted as a parallel phrase pair. A projected span in F described in Section 5.4.1 indicates that the span in F and its corresponding span in E are both continuous. If the projected span of F is transformed into non-continuous expressions in F' by pre-ordering, then a parallel phrase pair for the transformed expressions in F' cannot be extracted as a phrase pair. Therefore, it is thought to be optimal that F be reordered, as much as possible, into F' using the same method for pre-ordering input sentences so that this problem can be avoided.

Thus, we pre-order F in the training data into F' as follows. Partial syntactic structures are first projected onto the source language sentences in the training data using the method described in Section 5.4.1. The source language sentences are then parsed and reordered using the ITG parsing model as described in Section 5.5.1 within the constraints of the projected spans. When there are ambiguities in the projected spans, the laxest constraints are applied for each tree derivation.

5.6 Experiment

Our main target is the translation between widely different word orders, such as SOV and SVO, with a high quality target language parser. Therefore, we conducted Japanese-to-English (JE) translation as a case of the translation from an SOV language to an SVO language. In addition, we conducted Chinese-to-English (CE) translation as a case of the translation from an SVO language to another SVO language, which is more similar in word order than Japanese and English. We investigated the effectiveness of our method by comparing it with other methods. The patent data from the NTCIR-9 and NTCIR-10 Patent Machine Translation Tasks [Goto et al., 2011; Goto et al., 2013a] was used for the experiment.

5.6.1 Common Settings

The training data and the development data for NTCIR-9 and NTCIR-10 are the same, but the test data is different. There were approximately 3.18 million sentence pairs for the JE training data and 1 million sentence pairs for the CE training. The development data consists of 2,000 sentence pairs. There were 2,000 test sentences for NTCIR-9 and 2,300 for NTCIR-10. The reference data for each test sentence is a single reference translation.

We used Enju [Miyao and Tsujii, 2008] to parse the English sentences in the training data. We applied a parsing customization for patent sentences [Isozaki et al., 2012]. MeCab[4] was used for Japanese segmentation, and the Stanford segmenter[5] was used for Chinese segmentation. We adjusted the tokenization of alphanumeric characters in Japanese to be the same as for the English.

The translation model was trained using the sentences with lengths of 40 words or less and with English side sentences that could be parsed to produce binary syntactic tree structures. Approximately 2.06 million sentence pairs were used to train the translation model for JE. Approximately 0.40 million sentence pairs were used to train the translation model for CE. GIZA++ and grow-diag-

[4] `http://mecab.sourceforge.net/`

[5] `http://nlp.stanford.edu/software/segmenter.shtml`

final-and heuristics were used to obtain word alignments. In order to reduce word alignment errors, we removed articles {a, an, the} in English and particles {*ga*, *wo*, *wa*} in Japanese before performing word alignments because these function words do not have corresponding words in the other languages. After word alignment, we restored the removed words and shifted the word alignment positions to the original word positions.

We used 5-gram language models with modified Kneser-Ney discounting [Chen and Goodman, 1998] using SRILM [Stolcke et al., 2011]. The language models were trained using the English sentences from the bilingual training data.

The SMT weighting parameters were tuned by MERT [Och, 2003] using the development data. To stabilize the MERT results, we tuned the parameters three times by MERT using the first half of the development data and we selected the SMT weighting parameter set that performed the best on the second half of the development data based on the BLEU scores from the three SMT weighting parameter sets.

5.6.2 Training and Settings for the Proposed Method

Next is a description of how the proposed method (PROPOSED) was performed. As the training data of our pre-ordering model, source language full binary syntactic tree structures were produced for 0.1 million source language training sentences selected by the following process. The source language training sentences were sorted based on the coverage rates of the spans of the projected partial syntactic structures. We selected the top 0.1 million unique source language sentences.[6] To produce the probabilistic CFG model and the probabilistic model for the unsupervised POS tagging, we used the Gibbs sampler for 100 iterations. We used $|\mathcal{T}| = 50$, which is the same number of word classes used in the Moses default setting, where $|\mathcal{T}|$ is the number of POS tag types. The Berkeley parser [Petrov et al., 2006], which is an implementation of the state split probabilistic CFG based parser, was used to train our pre-ordering model and was used to parse using the pre-ordering model. We performed 6 split-merge iterations as the same iteration

[6]We did not conduct experiments using larger training data because there would have been a very high computational cost in building probabilistic models for parsing.

of the parsing model for English [Petrov et al., 2006]. The phrase-based SMT system Moses [Koehn et al., 2007] was used to translate from F' into E with a distortion limit of 6, which limited the number of words for word reordering to a maximum number.

5.6.3 Training and Settings for the Compared Methods

We used the following six comparison methods.

- Phrase-based SMT with lexicalized reordering models (PBMT$_L$) [Koehn et al., 2007]
- Hierarchical phrase-based SMT (HPBMT) [Chiang, 2007]
- String-to-tree syntax-based SMT (SBMT) [Hoang et al., 2009]
- Phrase-based SMT with a distortion model (PBMT$_D$) [Goto et al., 2014]
- Pre-ordering using a source language dependency parser (SRCDEP) [Genzel, 2010][7]
- Pre-ordering without requiring a parser (LADER) [Neubig et al., 2012][8]

We used Moses [Koehn et al., 2007; Hoang et al., 2009] for PBMT$_L$, HPBMT, SBMT, SRCDEP, and LADER. We used an in-house standard phrase-based SMT decoder compatible with the Moses decoder with a distortion model [Goto et al., 2014] for PBMT$_D$.

PBMT$_L$ used the MSD bidirectional lexicalized reordering models [Koehn et al., 2005] that were built using all of the data used to build the translation model.

The distortion models for PBMT$_D$ were trained using the last 0.2 million source language sentences used to build the translation model and their word alignments. This setting is the same as that of the experiments in [Goto et al.,

[7]There are three variations of metrics for selecting rules. We implemented variant 1 (optimizing crossing score), which achieved the best score for JE translation in the three variations in the experiments by Genzel [2010], and used that implementation.

[8]We used the lader implementation available at `http://www.phontron.com/lader/`.

2014]. PBMT$_D$ used source language POS tags produced by MeCab for Japanese and by the Stanford tagger[9] for Chinese.

SRCDEP requires a source language dependency parser. Therefore, a comparison of PROPOSED with SRCDEP is unfair and PROPOSED is at a disadvantage to SRCDEP. We used CaboCha[10] [Taku Kudo, 2002] and POS tags produced by MeCab to obtain Japanese dependency structures[11] and used the Stanford parser[12] and POS tags produced by the Stanford tagger to obtain Stanford dependencies for Chinese [Chang et al., 2009]. Note that there are publicly available Japanese dependency parsers but there are no publicly available Japanese constituency parsers. The pre-ordering rules of SRCDEP were built using all of the data used to build the translation model.

The pre-ordering models for LADER were trained using the same 0.1 million source language sentences and their word alignments as the training data for the pre-ordering models of PROPOSED. Source language word classes produced by the Moses tool kit were used. Note that while the LADER pre-ordering method does not use a parser, the training data for LADER was selected using a target language parser. We performed 100 iterations for training the LADER pre-ordering model.[13]

[9] `http://nlp.stanford.edu/software/tagger.shtml`

[10] `https://code.google.com/p/cabocha/`

[11] The CaboCha parser does not output word-based dependencies, but segment-based dependencies. Each segment, which is called a *bunsetsu*, is comprised of at least one content word with or without its following function words. We converted the segment-based dependencies to word-based dependencies as follows: When a punctuation mark is included in a segment, the segment is split into a segment without the punctuation mark and a segment that consists only of the punctuation mark. Each word except for the last word in a segment depends on (modifies) the right adjacent word. The last word in a segment depends on the headword of the parent (modified) segment. The headword in a segment is the last content word in the segment.

The CaboCha parser does not output dependency relations. We added dependency relations to the word-based dependencies as follows: When the last word in a segment is a particle, we used the particle as the dependency relation between the word and its patent (modified) word because particles are case markers in many cases in Japanese. For other words, we used "none" as their dependency relations to their parent words.

[12] `http://nlp.stanford.edu/software/lex-parser.shtml`

[13] We also tested 200 iterations for JE translation and found that the results with 200 iterations did not improve as compared to the results with 100 iterations.

Table 5.2: Japanese-English Evaluation Results

	Parser		Pre-	NTCIR-9		NTCIR-10	
	Source	Target	ordering	RIBES	BLEU	RIBES	BLEU
PBMT$_{L\text{-}4}$				65.48	26.73	65.53	27.44
PBMT$_{L\text{-}20}$				68.79	30.92	68.30	31.07
HPBMT				70.11	30.29	69.69	30.77
SBMT		✓		72.54	31.94	71.32	32.40
PBMT$_D$				73.54	33.14	72.23	33.87
SRCDEP	✓		✓	71.88	29.23	71.20	29.40
LADER			✓	74.31	32.98	73.98	33.90
PROPOSED		✓	✓	**76.35**	**33.83**	**75.81**	**34.90**

For PBMT$_L$, distortion limits of 4 or 20 were used for JE translation and distortion limits of 4 or 10 were used for CE translation. The reason for this is that 20 was the best for JE translation and 10 was the best for CE translation among 10, 20, 30, and ∞ in the experiments of [Goto et al., 2014] and Genzel [2010] used a baseline phrase-based SMT that was capable of local reordering of up to 4 words. To distinguish between the distortion limits for PBMT$_L$, we indicate a distortion limit as a subscript of PBMT$_L$, such as PBMT$_{L\text{-}20}$ for a distortion limit of 20. For PBMT$_D$, a distortion limit of 20 was used for JE translation and a distortion limit of 10 was used for CE translation. An unlimited max-chart-span was used for HPBMT and SBMT and a distortion limit of 6 was used for the pre-ordering methods of SRCDEP and LADER. The default values were used for the other system parameters.

5.6.4 Results and Discussion

We evaluated translation quality based on the case-insensitive automatic evaluation scores BLEU-4 [Papineni et al., 2002] and RIBES v1.01 [Isozaki et al., 2010a]. RIBES is an automatic evaluation measure based on word order correlation coefficients between reference sentences and translation outputs. Our main results for JE translation are presented in Table 5.2 and those for CE translation are

Table 5.3: Chinese-English Evaluation Results

	Parser		Pre-	NTCIR-9		NTCIR-10	
	Source	Target	ordering	RIBES	BLEU	RIBES	BLEU
PBMT$_{L\text{-}4}$				75.02	29.22	74.24	30.65
PBMT$_{L\text{-}10}$				76.11	31.20	75.41	32.34
HPBMT				77.68	32.39	77.45	33.61
SBMT		✓		78.44	32.47	77.68	33.90
PBMT$_D$				77.98	33.03	77.48	34.28
SRCDEP	✓		✓	76.88	28.85	76.14	29.36
LADER			✓	78.18	30.80	77.06	31.12
PROPOSED		✓	✓	**81.61**	**35.16**	**81.05**	**36.22**

presented in Table 5.3. In these tables, check marks in a column indicate usage for that method. Bold numbers indicate not being significantly lower than the best result (that is, non-bold numbers indicate being significantly lower than the best result) in each test set and in each evaluation measure using the bootstrap resampling test at a significance level $\alpha = 0.01$ [Koehn, 2004].[14]

PROPOSED achieved the best scores for both RIBES and BLEU in both the NTCIR-9 and NTCIR-10 data sets, and for both JE and CE translation. Since RIBES is sensitive to global word order and BLEU is sensitive to local word order, this confirmed the effectiveness of PROPOSED for both global and local word ordering.

Now to compare the effects of the differences in the approaches. First, we compare our method with three existing methods that do not use a parser and conduct word selection and reordering jointly. For both the NTCIR-9 and NTCIR-10 results for JE translation, the RIBES and BLEU scores for PROPOSED were higher than those for the standard phrase-based SMT (PBMT$_{L\text{-}20}$), the hierarchical phrase-based SMT (HPBMT), and the phrase-based SMT with a recent distortion model (PBMT$_D$). These results confirmed that pre-ordering was effective compared to these methods that do not use a parser and conduct word selection and

[14]We used this indication method because this method can indicate simply the results of hypothesis test for one result and many baseline results.

reordering jointly for JE patent translation. The tendencies of the CE translation results were the same as those of the JE translation results. These results confirmed that pre-ordering was also effective for CE patent translation.

Next, we compared our method with an existing method that uses a target language syntactic parser, SBMT. The required resources are the same as those for PROPOSED. For both the NTCIR-9 and NTCIR-10 results for JE translation, the RIBES and BLEU scores for PROPOSED were higher than those for the string-to-tree syntax-based SMT (SBMT). These results confirmed that pre-ordering was effective compared to the method that utilizes target language syntactic structures and conducts word selection and reordering jointly for JE patent translation. The tendencies of the CE translation results were also the same as those of the JE translation results.

We then compared our method with an existing method using a source language dependency parser, SRCDEP. For both the NTCIR-9 and NTCIR-10 results for JE translation, the RIBES and BLEU scores for PROPOSED were higher than those for SRCDEP. These results confirmed that our method was effective compared to a method using a source language dependency parser [Genzel, 2010] for JE patent translation. The tendenies of the CE translation results were also the same as those of the JE translation results.

Here, we confirm the effects of SRCDEP for JE (that is, between SOV and SVO) translation.[15] SRCDEP produced BLEU scores that were about 2 BLEU points higher than those for PBMT$_{L\text{-}4}$. These results were consistent with the experiment results of Genzel [2010]. Genzel [2010] compared their method with their baseline phrase-based SMT that was capable of local reordering of up to 4 words. Although SRCDEP produced better BLEU scores than those for PBMT$_{L\text{-}4}$ and better RIBES scores than those for PBMT$_{L\text{-}4}$ and PBMT$_{L\text{-}20}$, the BLEU scores for SRCDEP were lower than those for PBMT$_{L\text{-}20}$. This indicates that even if a source language dependency parser is used, it is not easy to improve JE translation quality by pre-ordering.[16] One of the reasons that SRCDEP was unable to achieve

[15]Since Genzel [2010] reported the translation results from English (an SVO language) to SOV or VSO languages including Japanese and did not report the translation results between English and Chinese (an SVO language to an SVO language), we discuss SRCDEP for JE translation.

[16]There were also systems that were pre-ordering methods using a source language dependency

scores on par with PROPOSED is thought to be because when SRCDEP changes the order of child nodes, the reordering rules consider only the local information. Reordering, however, should consider sentence level consistency. For example, an SOV sentence in Japanese should be reordered into an SVO sentence for JE translation. However, when the subject in a sentence is omitted in Japanese, an OV sentence in Japanese should not be reordered into a VO sentence. This is because such sentences are usually translated into sentences in the passive voice and the objects in Japanese become subjects in the translated sentences. Since SRCDEP pre-ordering rules only consider local information, a rule is unable to handle the difference between SOV and OV when the rule does not consider S, such as when swapping O and V. In contrast, PROPOSED considers sentence level consistency.

Finally, we compared our method with an existing pre-ordering method that does not use a syntactic parser, LADER. For both the NTCIR-9 and NTCIR-10 results for JE translation, the RIBES and BLEU scores for PROPOSED were higher than those for LADER. [17] These results confirmed that utilizing syntactic struc-

parser in the Japanese-to-English translation subtasks at NTCIR-10 and NTCIR-7.

At NTCIR-10, there was one system (name: JEPREORDER, ID: NTITI-je-2) that was a source syntax-based pre-ordering method using manually produced pre-ordering rules and a Japanese dependency parser with a case structure analyzer [Sudoh et al., 2013]. Compared with the baseline hierarchical phrase-based SMT system (ID: BASELINE1-1) at NTCIR-10, the BLEU score for JEPREORDER was higher than that of the baseline system, but the RIBES score was not better than that of the baseline system in Table 1 in [Sudoh et al., 2013].

At NTCIR-7, there was one system (ID: MIT (2)) that was a source syntax-based pre-ordering method using manually produced pre-ordering rules and a Japanese dependency parser [Katz-Brown and Collins, 2008]. The system was unable to produce a BLEU score that was better than that of the baseline phrase-based SMT system at NTCIR-7.

[17] Note that although LADER works without a syntactic parser, the scores for LADER in Table 5.2 could not be achieved without a syntactic parser because a syntactic parser was used in the selection process of the training data for the pre-ordering model for LADER. When the last 0.1 million source language sentences of the training data were used as the training data for the pre-ordering model of LADER for JE translation, the RIBES scores for NTCIR-9 and NTCIR-10 were 72.33 and 70.96 respectively, and the BLEU scores for NTCIR-9 and NTCIR-10 were 32.30 and 33.07 respectively. We used the same training data for the pre-ordering model of LADER as the training data for the pre-ordering model of PROPOSED to perform a fair comparison with PROPOSED.

tures for pre-ordering was effective compared to not utilizing syntactic structures in JE patent translation.[18] The tendencies of the CE translation results were also the same as those of the JE translation results.

We checked the average coverage rates of the projected spans except for the sentence root spans.[19] The coverage rates for each source language sentence were calculated by dividing the number of projected spans except for the sentence root spans by the number of words in the sentence minus two.[20] The average coverage rates for the data used to build the translation model were 0.562 for Japanese and 0.601 for Chinese. The average coverage rates for the 0.1 million sentences used to produce full binary tree structures were 0.856 for Japanese and 0.828 for Chinese. With these projected partial structures, full binary tree structures were produced using the methods described in Sections 5.4.2, 5.4.3, and 5.5.2.[21]

In these experiments, we did not compare our method to post-ordering methods. However, for the same NTCIR-9 test data, the RIBES and BLEU scores for PROPOSED were higher than the RIBES and BLEU scores for a post-ordering method in the experiments in [Goto et al., 2013b], which uses the same state split probabilistic CFG method for the ITG parsing model as our method did for the ITG parsing model. In addition, PROPOSED has an advantage over the post-ordering methods of [Sudoh et al., 2011b; Goto et al., 2013b; Hayashi et al., 2013]. These post-ordering methods use manually defined high quality pre-ordering rules

[18]There was also a system that was a pre-ordering method without requiring a parser in the Japanese-to-English translation subtask at NTCIR-9. The system of the NAIST group [Kondo et al., 2011] used a pre-ordering method [Tromble and Eisner, 2009] that learned a pre-ordering model automatically without requiring a parser. The system was unable to produce a BLEU score that was better than those for the baseline systems of phrase-based SMT and hierarchical phrase-based SMT at NTCIR-9, although it could produce a RIBES score that was better than those for the baseline systems.

[19]Since sentence root spans are obvious and do not need to be projected, we did not include the sentence root spans to calculate the coverage rates.

[20]The number of brackets in a full binary tree is the number of words in a sentence minus one. We subtract one from the number of brackets for removing the sentence root brackets.

[21]It does not mean that all of the projected spans are included in the full binary trees. The reason is explained in Section 5.4.3.

of head-finalization from English to Japanese [Isozaki et al., 2012], so it is not easy to apply these methods to other language pairs. In contrast, PROPOSED does not require these manually defined rules, and so could be applied to other languages.

5.6.5 Evaluation Focusing on Projection

To investigate the effects of our projection method, we compared the parsing quality by our method with that by the method of Jiang et al. [2011]. Following the previous work, we used the same FBIS Chinese-English parallel corpus (LDC2003E14) as [Jiang et al., 2011] used, which consists of 0.24 million sentence pairs, to obtain projected constituent structures and evaluated our projected parser on the same test data that is the subset of Chinese Treebank 5.0 (CTB 5.0; LDC2005T01), which consists of no more than 40 words after the removal of punctuations, just as [Jiang et al., 2011] did.

Following the previous work, we used the same evaluation metric of unlabeled F_1 as Jiang et al. [2011] used, which is the harmonic mean of the unlabeled precision and recall, which was defined by Klein [2005] (pp.19–22). The evaluation for unlabeled brackets differs slightly from the standard PARSEVAL metrics: multiplicity of brackets is ignored, brackets of span one are ignored, and bracket labels are ignored. Previous research of [Jiang et al., 2011] and [Klein, 2005] (p.16) removed punctuations before the evaluation. We followed this by removing words that have punctuation tags of PU in CTB 5.0 after parsing.

We used our method described in Sections 5.4.1 to 5.4.3 and 5.6.2 to obtain projected constituent structures. To reduce computational costs, we changed one of the settings described in Section 5.6.2. For the projection, we selected the top 50 thousand unique source language sentences on the basis of the coverage rates of the spans of the projected partial syntactic structures from the FBIS corpus, whereas we selected the top 0.1 million unique source language sentences in Section 5.6.2.[22] The average coverage rate of the projected spans except for the sentence root spans for the 50 thousand sentences used to produce full binary

[22]The average span coverage rate of the top 0.1 million sentences of the FBIS corpus was lower than that of the NTCIR-9/10 data. A lower rate increases ambiguities of parse trees and computational costs.

Table 5.4: Evaluation Results on Parsing

	F_1 (CTB5-40)
[Jiang et al., 2011]	49.2*
Proposed method	**56.1**

Note: $*$ denotes "not our experiment."

tree structures was 0.795. The Berkeley parser, which was also used by Jiang et al. [2011] for the same purpose, was used to build the parsing model from the projected constituent structures and to parse the test data.

Jiang et al. [2011] used the gold POS tags of CTB 5.0 for parsing and a supervised Chinese POS tagger for tagging the FBIS corpus. In contrast, our method did not use the gold POS tags of CTB 5.0 or a supervised Chinese POS tagger. Therefore, a comparison of our method with that of Jiang et al. [2011] is unfair since our method is at a disadvantage to theirs.

The evaluation results are given in Table 5.4. Although our method does not require source language POS tags, our method produced a F_1 higher than that of Jiang et al. [2011]. This confirmed the effectiveness of our projection method.

5.7 Summary

We have presented a pre-ordering method that uses a target language parser to utilize syntactic structures without requiring a source language parser. In order to produce our ITG-based pre-ordering model utilizing syntactic phrase labels, our method projects the target language constituent structures of the target language training sentences onto their corresponding source language sentences and produces a probabilistic CFG model and a probabilistic model for unsupervised POS tagging using the Pitman-Yor process for parsing to produce full binary constituent structures for the source language training sentences. In the experiments on Japanese-to-English and Chinese-to-English patent translation, our method achieved a significant improvement in translation quality as measured by both RIBES and BLEU over phrase-based SMT, hierarchical phrase-based SMT,

string-to-tree syntax-based SMT, an existing pre-ordering method without using a parser, and an existing pre-ordering method using a source language dependency parser. Since RIBES is sensitive to global word order and BLEU is sensitive to local word order, we concluded that the proposed method was better than the compared methods at global and local word ordering. We also confirmed the effectiveness of our projection method for constituent structures compared with an existing projection method for constituent structures using the FBIS corpus and Chinese Treebank 5.0. Future work will involve cooperating with a source language parser when one is available.

Chapter 6

Comparison of the Proposed Methods

In this chapter, we compare the three proposed reordering methods: the phrase-based SMT with the proposed distortion model (DISTORTION) described in Chapter 3, the proposed post-ordering method (POST-ORDERING)[1] described in Chapter 4, and the proposed pre-ordering method (PRE-ORDERING) described in Chapter 5.

6.1 Applicability Comparison

First, we will compare the applicability of the three proposed methods. The applicability of each model is summarized in Table 6.1. DISTORTION does not require a parser, so it is applicable to any language pair. POST-ORDERING uses a target language binary constituency parser, and manually defined pre-ordering rules of head-finalization [Isozaki et al., 2012], which reorder words in a sentence into a head-final word order. POST-ORDERING is applicable for translations from head-final languages to languages with parsers, such as translations from Japanese to English. PRE-ORDERING uses a target language constituency parser. PRE-ORDERING produces pre-ordering rules automatically. Therefore, PRE-ORDERING is applicable to translations from any language into any language with parsers.

[1]The proposed post-ordering method that does not use source language syntax.

Table 6.1: Applicability of the Proposed Methods to Languages

	Source	Target
DISTORTION (Chapter 3)	Any language	Any language
POST-ORDERING (Chapter 4)	Head-final language	Languages with parsers
PRE-ORDERING (Chapter 5)	Any language	Languages with parsers

Table 6.2: Evaluation Results for NTCIR-9 Japanese-English Translation

	RIBES	BLEU
DISTORTION	73.54	33.14
POST-ORDERING	74.85	32.15
PRE-ORDERING	76.35	33.83

6.2 Comparison of Translation Quality

We will now compare translation quality for a Japanese-to-English patent translation among the three proposed methods. To enable a fair comparison, we conducted a POST-ORDERING experiment using a set of training data that was smaller than the set used in the experiments in Chapter 4. To produce the translation model for POST-ORDERING, we used the training sentences that were 40 words or less in length and that had English side sentences that could be parsed to produce binary syntactic tree structures. The methods for selecting the training data were the same as those for DISTORTION (PBMT_D) and PRE-ORDERING (PROPOSED) in Table 5.2. We randomly selected 0.2 million sentences as the training data for the ITG parsing model of POST-ORDERING. This data was the same size as the training data used for the proposed distortion model of DISTORTION (PBMT_D) in Table 5.2.

The evaluation results for the NTCIR-9 test data are summarized in Table 6.2. The scores for DISTORTION (PBMT_D) and PRE-ORDERING (PROPOSED) were taken from Table 5.2. As shown in Table 6.2, both the RIBES and BLEU scores for PRE-

ordering were higher than those for Distortion and Post-ordering. This confirmed that Pre-ordering was the most effective among the three methods.

There are two main reasons why Pre-ordering was better than Post-ordering. The first is that reordering by Post-ordering was affected by word selection errors because Post-ordering conducts reordering after word selection. In contrast, reordering by Pre-ordering does not suffer from word selection errors because Pre-ordering conducts reordering before word selection. The second reason is as follows. In some cases, correct reorderings are prevented by the ITG constraints. In such cases, it is thought that Pre-ordering works more reliably than Post-ordering. Training data for the ITG parsing model of Pre-ordering can include these cases. Therefore, although the ITG parsing model of Pre-ordering cannot produce correct reorderings in these cases, the ITG parsing model of Pre-ordering is expected to produce the best reordering given the ITG constraints. In contrast, training data for the ITG parsing model of Post-ordering does not include such cases. Thus, for these cases, the ITG parsing model of Post-ordering cannot be relied upon for stable analysis. In addition to the two main reasons mentioned above, when correct reorderings are not produced for these cases at the reordering stage of Pre-ordering, Pre-ordering may produce correct reorderings at the word selection stage, because small reorderings are permitted at this stage. In contrast, Post-ordering cannot produce correct reorderings for these cases.

The RIBES score for Post-ordering was higher than that for Distortion. However, the BLEU score for Distortion was higher than that for Post-ordering. Therefore, it is difficult to compare between Post-ordering and Distortion. Since RIBES scores have a higher correlation with human evaluation than BLEU scores for Japanese-to-English translation at NTCIR-9 [Goto et al., 2011] and NTCIR-10 [Goto et al., 2013a], we propose that Post-ordering is superior to Distortion for Japanese-to-English patent translation.

6.3 Characteristics of the Proposed Methods

Here, we list the characteristics of the proposed methods other than those described above. The proposed distortion model of DISTORTION is an essential components of phrase-based SMT. When phrase-based SMT is used and reordering is needed, the proposed distortion model always contributes to translation quality.

POST-ORDERING can output target language syntactic structures that are produced by parsing and syntactic transfer using ITG, but DISTORTION and PRE-ORDERING do not output target language syntactic structures. Therefore, when applications that use translation outputs require the syntactic structures of translation outputs, POST-ORDERING is useful.

Chapter 7

Conclusion

Statistical machine translation is a promising technology that is rapidly evolving. In this thesis, we discussed reordering, which is one of the main elements of SMT. We proposed three methods for modeling structural differences between languages to improve reordering in SMT when certain restrictions exist with respect to the availability of parsers.

7.1 Summary

Chapter 1 introduced the history of machine translation research. We then described current issues in statistical machine translation. Specifically, we discussed the challenges of translating between languages with largely different word orders. We also stated the objectives of this thesis, and gave an overview of our approaches.

Chapter 2 introduced the SMT framework, reordering methods, and evaluation methods.

Chapter 3 proposed a new distortion model for phrase-based SMT for cases in which both a source language parser and a target language parser are unavailable. During the translation process, a distortion model estimates the source word position to be translated next (subsequent position; SP) given the last translated source word position (current position; CP). The SP depends on structural differences between languages. In previous methods, an SP candidate is identified by

its distance and direction from the CP. Distances and directions between the CP and SP candidates are modeled and the probabilities for each distance and direction are calculated. When probabilities for each distance are modeled, training data corresponding to each distance become very small; that is, the data scarcity occurs. Therefore, existing methods calculate probabilities for each distance class. However, the probabilities for distances belonging to the same distance class are the same. Therefore, such models are not able to appropriately model structural differences between languages. In contrast, the proposed distortion model can approximately model structural differences between languages without a parser. Our model uses label sequences that can characterize elements of syntax, such as VP or NP. It directly calculates the probabilities for each SP candidate being the SP without mediating distances to identify each SP candidate. The proposed distortion model can simultaneously consider the word at the CP, the word at an SP candidate, the context of the CP and an SP candidate, relative word order among the SP candidates, and the words between the CP and an SP candidate. These considered elements are called *rich context*. Our model considers rich context and structural differences between languages by discriminating label sequences that specify spans from the CP to each SP candidate. This enables our model to learn the effect of relative word order among SP candidates as well as to learn the effect of distances from the training data. In contrast to the learning strategy used by existing methods, our learning strategy is novel in that the model learns preference relations among the features of SP candidates in each sentence of the training data. This learning strategy enables consideration of all the rich contexts simultaneously. In our experiments, our model had higher BLEU and RIBES scores for Japanese-English, Chinese-English, and German-English translation than the lexical reordering models using NTCIR-9 Patent Machine Translation Task data [Goto et al., 2011], NIST 2008 Open MT task data, and WMT 2008 Europarl data [Callison-Burch et al., 2008].

Chapter 4 proposed a post-ordering method that reorders target words by parsing for Japanese-to-English translation using a target language binary constituency parser. We employed the post-ordering framework and improved upon its reordering method. A previously proposed post-ordering method for Japanese-

to-English translation first translates a source language sentence into a HFE sentence, which is a sequence of target language words in a word order similar to that of the source language, then reorders the HFE sentence into a target language word order using phrase-based SMT. The previous method conducted reordering without using syntactic structures. In contrast, in our method, the HFE sentence is reordered by: (1) parsing the HFE sentence using an ITG parsing model that uses syntactic categories to obtain syntactic structures, which are similar to the syntactic structures of the source language, and (2) transferring the obtained syntactic structures into target language syntactic structures according to the ITG. We modeled structural differences between HFE and the target language (English) as an ITG parsing model. Our method is the first post-ordering method that conducts reordering based on parsing and ITG. We formulated a method for modeling differences between Japanese syntactic structures and the corresponding English structures that was easily manageable compared with previous methods. It is easier to model differences between HFE syntactic structures and the corresponding English syntactic structures than modeling differences between Japanese syntactic structures and the corresponding English syntactic structures. This is because HFE syntactic structures are perfectly synchronized with the corresponding English syntactic structures, whereas in many cases, some elements of Japanese syntactic structures are not synchronized with the corresponding English syntactic structures. We conducted experiments using Japanese-to-English patent translation using the patent data from the NTCIR-9 and NTCIR-8 Japanese-to-English Patent Machine Translation subtasks [Goto et al., 2011; Fujii et al., 2010]. In the experiments, our method outperformed phrase-based SMT, hierarchical phrase-based SMT, string-to-tree syntax-based SMT, and post-ordering methods based on SMT for both RIBES and BLEU.

Chapter 5 proposed a pre-ordering method that uses a target language parser to process syntactic structures without a source language parser. Since conventional pre-ordering methods usually use a source language syntactic parser to change the source language word order, these methods cannot be applied when source language syntactic parsers are unavailable. Several pre-ordering methods exist that change the source language word order using BTG, which does not use

syntax, without requiring a parser. In contrast, the proposed method uses an ITG parsing model that uses syntactic categories to change a source language word order. We modeled structural differences between languages as an ITG parsing model for a source language that uses syntactic categories of the target language syntax. We reordered source language sentences by parsing using the ITG parsing model. To train the ITG parsing model, we produced syntactic constituent structures of source language training sentences by (1) projecting the constituent structures of target language sentences to the corresponding source language sentences, (2) producing a probabilistic CFG model and a probabilistic model for unsupervised part-of-speech tagging using the projected partial structures and the Pitman-Yor process, and (3) producing full binary syntactic structures within the constraints of the projected partial structures by parsing using the probabilistic models. The ITG parsing model for the source language was built using the produced source language binary syntactic constituent structures. The main contributions are summarized as follows: (i) We proposed a new effective pre-ordering framework that can process syntactic structures using a target language syntactic parser without a source language syntactic parser. (ii) We formulated a method for modeling differences between syntactic structures of languages that is easy to manage. It is easier to model differences between projected syntactic structures and the corresponding target language syntactic structures than to model differences between non-projected source language syntactic structures and the corresponding target language syntactic structures. This is because the level of synchrony between the projected syntactic structures and the corresponding target language syntactic structures is higher than that between the non-projected source language syntactic structures and the corresponding target language syntactic structures, as the projection produces source language syntactic structures that are maximally synchronized with the corresponding target language syntactic structures. Experiments on Japanese-English and Chinese-English patent translation indicated that our method outperformed string-to-tree syntax-based SMT, an existing pre-ordering method that does not use a parser, and an existing pre-ordering method that uses a source language dependency parser. This was achieved using patent data from the NTCIR-9 and NTCIR-10 Patent Machine

Translation Tasks [Goto et al., 2011; Goto et al., 2013a].

Chapter 6 compared the proposed reordering methods. The distortion model described in Chapter 3 does not require a parser, so it is applicable to any language pair. The post-ordering method described in Chapter 4 can be applied to translations from head-final languages to languages with parsers such as Japanese to English because the post-ordering method uses the head-finalization pre-ordering rules. This method can output target language syntactic structures. The pre-ordering method described in Chapter 5 can be applied to translations from any language into languages with parsers. This method achieved the best translation quality among the three proposed methods for Japanese-to-English patent translation.

The three proposed methods for modeling structural differences between languages were significantly more effective than the existing methods that we compared. We concluded that our modeling methods were very effective for improving the quality of translation between languages with largely different word orders. It is difficult to always achieve perfect translation using any single translation method. Therefore, it is important to have access to two or more different translation methods, because all the outputs from different methods may be used or the best output can be selected from the outputs using a system combination method.

7.2 Future Work

Several problems remain, necessitating future research.

For cases in which a parser is not available: Although syntactic structures cannot be used, it is important to use tree structures for reordering. Therefore, future work should focus on phrase-based SMT methods that use both the proposed distortion model and the BTG constraints. Future research should also include possible ways to use the proposed distortion model in hierarchical phrase-based SMT.

For cases in which a parser is available: There are correct reorderings that

the ITG constraints cannot produce for certain input sentences. For example, the fundamental word order in Japanese is SOV, but a word order of OSV is also acceptable in Japanese. However, our post-ordering and pre-ordering methods can only reorder based on ITG constraints. A sentence that has an OSV word order cannot be transferred into (S (V O)) by ITG because O and V are not continuous. In this case, it is necessary to convert an input sentence that has an OSV word order into a sentence that has an SOV word order at the pre-processing stage, prior to the translation by our post-ordering or pre-ordering methods. Future work will address this problem. This problem will be solved by introducing a pre-process that conducts reordering based on dependency structures. Correct syntactic structures are important for correct reordering. Thus, improving parsing accuracy is also an important topic for future work. It is difficult to ensure correct parsing, so robust translation methods for parsing errors are important topics for future research. One simple method for realizing robust translation is a system combination technique, in which outputs are selected from translations produced by different methods, including methods that use different parsers and those that do not use a parser.

Bibliography

[Aho and Ullman, 1969] A. V. Aho and J. D. Ullman. 1969. Syntax directed translations and the pushdown assembler. *J. Comput. Syst. Sci.*, 3(1):37–56, February.

[Aho and Ullman, 1972] Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation, and Compiling.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[Aikawa and Ruopp, 2009] Takako Aikawa and Achim Ruopp. 2009. Chained system: A linear combination of different types of statistical machine translation systems. In *Proceedings of the twelfth Machine Translation Summit*, Ottawa, Ontario, Canada, August. International Association for Machine Translation.

[Al-Onaizan and Papineni, 2006] Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 529–536, Sydney, Australia, July. Association for Computational Linguistics.

[Badr et al., 2009] Ibrahim Badr, Rabih Zbib, and James Glass. 2009. Syntactic phrase reordering for English-to-Arabic statistical machine translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 86–93, Athens, Greece, March. Association for Computational Linguistics.

[Berger et al., 1996] Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, March.

[Blunsom et al., 2009] Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 782–790, Suntec, Singapore, August. Association for Computational Linguistics.

[Brown et al., 1993] Peter E. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

[Callison-Burch et al., 2008] Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical*

Machine Translation, pages 70–106, Columbus, Ohio, June. Association for Computational Linguistics.

[Chang et al., 2009] Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009. Discriminative reordering with Chinese grammatical relations features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation (SSST-3) at NAACL HLT 2009*, pages 51–59, Boulder, Colorado, June. Association for Computational Linguistics.

[Chen and Goodman, 1998] Stanley F. Chen and Joshua T. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Computer Science Group, Harvard University.

[Chen and Rosenfeld, 1999] Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical Report CMU-CS-99-108, Carnegie Mellon University.

[Chen et al., 2009] Han-Bin Chen, Jian-Cheng Wu, and Jason S. Chang. 2009. Learning bilingual linguistic reordering model for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 254–262, Boulder, Colorado, June. Association for Computational Linguistics.

[Cherry and Lin, 2007] Colin Cherry and Dekang Lin. 2007. Inversion transduction grammar for joint phrasal translation modeling. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 17–24, Rochester, New York, April. Association for Computational Linguistics.

[Cherry et al., 2012] Colin Cherry, Robert C. Moore, and Chris Quirk. 2012. On hierarchical re-ordering and permutation parsing for phrase-based decoding. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 200–209, Montréal, Canada, June. Association for Computational Linguistics.

[Cherry, 2013] Colin Cherry. 2013. Improved reordering for phrase-based translation using sparse features. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 22–31, Atlanta, Georgia, June. Association for Computational Linguistics.

[Chiang, 2007] David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

[Chiang, 2010] David Chiang. 2010. Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1443–1452, Uppsala, Sweden, July. Association for Computational Linguistics.

[Collins et al., 2005] Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan, June. Association for Computational Linguistics.

[DeNero and Uszkoreit, 2011] John DeNero and Jakob Uszkoreit. 2011. Inducing sentence structure from parallel corpora for reordering. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

[Ding and Palmer, 2005] Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 541–548, Ann Arbor, Michigan, June. Association for Computational Linguistics.

[Dugast et al., 2007] Loïc Dugast, Jean Senellart, and Philipp Koehn. 2007. Statistical post-editing on SYSTRAN's rule-based translation system. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 220–223, Prague, Czech Republic, June. Association for Computational Linguistics.

[Dyer and Resnik, 2010] Chris Dyer and Philip Resnik. 2010. Context-free reordering, finite-state translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 858–866, Los Angeles, California, June. Association for Computational Linguistics.

[Ehara, 2007] Terumasa Ehara. 2007. Rule based machine translation combined with statistical post editor for Japanese to English patent translation. In *Proceedings of MT Summit XI Workshop on Patent Translation*, pages 13–18, Copenhagen, Denmark, September. International Association for Machine Translation.

[Evgniou and Pontil, 2002] Theodoros Evgniou and Massimiliano Pontil. 2002. Learning preference relations from data. *Neural Nets Lecture Notes in Computer Science*, 2486:23–32.

[Feng et al., 2010] Yang Feng, Haitao Mi, Yang Liu, and Qun Liu. 2010. An efficient shift-reduce decoding algorithm for phrased-based machine translation. In *Coling 2010: Posters*, pages 285–293, Beijing, China, August. Coling 2010 Organizing Committee.

[Feng et al., 2013] Minwei Feng, Jan-Thorsten Peter, and Hermann Ney. 2013. Advancements in reordering models for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 322–332, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Fujii et al., 2010] Atsushi Fujii, Masao Utiyama, Mikio Yamamoto, Takehito Utsuro, Terumasa Ehara, Hiroshi Echizen-ya, and Sayori Shimohata. 2010. Overview of the patent translation task at the NTCIR-8 workshop. In *Proceedings of NTCIR-8*, pages 371–376.

[Galley and Manning, 2008] Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Honolulu, Hawaii, October. Association for Computational Linguistics.

[Galley et al., 2004] Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

[Ge, 2010] Niyu Ge. 2010. A direct syntax-driven reordering model for phrase-based machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 849–857, Los Angeles, California, June. Association for Computational Linguistics.

[Genzel, 2010] Dmitriy Genzel. 2010. Automatically learning source-side reordering rules for large scale machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 376–384, Beijing, China, August. Coling 2010 Organizing Committee.

[Goto et al., 2011] Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K. Tsou. 2011. Overview of the patent machine translation task at the NTCIR-9 workshop. In *Proceedings of NTCIR-9*, pages 559–578, December.

[Goto et al., 2013a] Isao Goto, Ka Po Chow, Bin Lu, Eiichiro Sumita, and Benjamin K. Tsou. 2013a. Overview of the patent machine translation task at the NTCIR-10 workshop. In *Proceedings of NTCIR-10*, pages 260–286, June.

[Goto et al., 2013b] Isao Goto, Masao Utiyama, and Eiichiro Sumita. 2013b. Post-ordering by parsing with ITG for Japanese-English statistical machine translation. *ACM Transactions on Asian Language Information Processing (TALIP)*, 12(4):17:1–17:22, October.

[Goto et al., 2014] Isao Goto, Masao Utiyama, Eiichiro Sumita, Akihiro Tamura, and Sadao Kurohashi. 2014. Distortion model based on word sequence labeling for statistical machine translation. *ACM Transactions on Asian Language Information Processing (TALIP)*, 13(1):2:1–2:21, February.

[Green et al., 2010] Spence Green, Michel Galley, and Christopher D. Manning. 2010. Improved models of distortion cost for statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 867–875, Los Angeles, California, June. Association for Computational Linguistics.

[Habash, 2007] Nizar Habash. 2007. Syntactic preprocessing for statistical machine translation. In *Proceedings of Machine Translation Summit XI*, pages 215–222.

[Hayashi et al., 2013] Katsuhiko Hayashi, Katsuhito Sudoh, Hajime Tsukada, Jun Suzuki, and Masaaki Nagata. 2013. Shift-reduce word reordering for machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1382–1386, Seattle, Washington, USA, October. Association for Computational Linguistics.

[He et al., 2010] Yanqing He, Yu Zhou, Chengqing Zong, and Huilin Wang. 2010. A novel reordering model based on multi-layer phrase for statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 447–455, Beijing, China, August. Coling 2010 Organizing Committee.

[Hoang et al., 2009] Hieu Hoang, Philipp Koehn, and Adam Lopez. 2009. A unified framework for phrase-based, hierarchical, and syntax-based statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 152–159.

[Hoshino et al., 2013] Sho Hoshino, Yusuke Miyao, Katsuhito Sudoh, and Masaaki Nagata. 2013. Two-stage pre-ordering for Japanese-to-English statistical machine translation. In *Proceedings of the Sixth International Joint Conference*

on Natural Language Processing, pages 1062–1066, Nagoya, Japan, October. Asian Federation of Natural Language Processing.

[Huang et al., 2006] Liang Huang, Kevin Knight, and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. In *Proceedings of the 7th Conference of the Association for Machine Translation of the Americas*, pages 66–73.

[Hwa et al., 2005] Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*, 11(3):311–325, September.

[Isozaki et al., 2010a] Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010a. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952.

[Isozaki et al., 2010b] Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010b. Head finalization: A simple reordering rule for SOV languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 244–251, Uppsala, Sweden, July. Association for Computational Linguistics.

[Isozaki et al., 2012] Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2012. HPSG-based preprocessing for English-to-Japanese translation. *ACM Transactions on Asian Language Information Processing*, 11(3):8:1–8:16, September.

[Jiang et al., 2011] Wenbin Jiang, Qun Liu, and Yajuan Lv. 2011. Relaxed cross-lingual projection of constituent syntax. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1201, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

[Katz-Brown and Collins, 2008] Jason Katz-Brown and Michael Collins. 2008. Syntactic reordering in preprocessing for Japanese → English translation: MIT system description for NTCIR-7 patent translation task. In *Proceedings of NTCIR-7*, pages 409–414.

[Khapra et al., 2013] Mitesh M. Khapra, Ananthakrishnan Ramanathan, and Karthik Visweswariah. 2013. Improving reordering performance using higher order and structural features. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 315–324, Atlanta, Georgia, June. Association for Computational Linguistics.

[Klein, 2005] Dan Klein. 2005. *The Unsupervised Learning of Natural Language Structure*. Ph.D. thesis, Stanford University.

[Koehn et al., 2003a] Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003a. Statistical phrase-based translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

[Koehn et al., 2003b] Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003b. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54.

[Koehn et al., 2005] Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*.

[Koehn et al., 2007] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

[Koehn, 2004] Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

[Koehn, 2010] Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

[Kondo et al., 2011] Shuhei Kondo, Mamoru Komachi, Yuji Matsumoto, Katsuhito Sudoh, Kevin Duh, and Hajime Tsukada. 2011. Learning of linear ordering problems and its application to J-E patent translation in NTCIR-9 PatentMT. In *Proceedings of NTCIR-9*, pages 641–645.

[Lafferty et al., 2001] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Li et al., 2007] Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou, and Yi Guan. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 720–727, Prague, Czech Republic, June. Association for Computational Linguistics.

[Liu and Nocedal, 1989] D.C. Liu and J. Nocedal. 1989. On the limited memory method for large scale optimization. *Mathematical Programming B*, 45(3):503–528.

[Liu et al., 2006] Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616, Sydney, Australia, July. Association for Computational Linguistics.

[Liu et al., 2009] Yang Liu, Yajuan Lü, and Qun Liu. 2009. Improving tree-to-tree translation with packed forests. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 558–566, Suntec, Singapore, August. Association for Computational Linguistics.

[Matusov et al., 2005] E. Matusov, S. Kanthak, and Hermann Ney. 2005. On the integration of speech recognition and statistical machine translation. In *Proceedings of Interspeech*, pages 3177–3180.

[Mehay and Brew, 2012] Dennis Nolan Mehay and Christopher Hardie Brew. 2012. CCG syntactic reordering models for phrase-based machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 210–221, Montréal, Canada, June. Association for Computational Linguistics.

[Miyao and Tsujii, 2008] Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic HPSG parsing. In *Computational Linguistics, Volume 34, Number 1*, pages 81–88.

[Nagao, 1984] Makoto Nagao. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In *Artificial and human intelligence: edited review papers presented at the international NATO Symposium, October 1981*, pages 173–180, Lyons, France.

[Neubig et al., 2011] Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. 2011. An unsupervised model for joint phrase alignment and extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*,

pages 632–641, Portland, Oregon, USA, June. Association for Computational Linguistics.

[Neubig et al., 2012] Graham Neubig, Taro Watanabe, and Shinsuke Mori. 2012. Inducing a discriminative parser to optimize machine translation reordering. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 843–853, Jeju Island, Korea, July. Association for Computational Linguistics.

[Ni et al., 2009] Yizhao Ni, Craig Saunders, Sandor Szedmak, and Mahesan Niranjan. 2009. Handling phrase reorderings for machine translation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 241–244, Suntec, Singapore, August. Association for Computational Linguistics.

[Och, 2003] Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July. Association for Computational Linguistics.

[Papineni et al., 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

[Petrov et al., 2006] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July. Association for Computational Linguistics.

[Petrov et al., 2008] Slav Petrov, Aria Haghighi, and Dan Klein. 2008. Coarse-to-fine syntactic machine translation using language projections. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 108–116, Honolulu, Hawaii, October. Association for Computational Linguistics.

[Petrov, 2010] Slav Petrov. 2010. Products of random latent variable grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Los Angeles, California, June. Association for Computational Linguistics.

[Pitman and Yor, 1997] Jim Pitman and Marc Yor. 1997. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900.

[Quirk et al., 2005] Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 271–279, Ann Arbor, Michigan, June. Association for Computational Linguistics.

[Ramanathan et al., 2008] Ananthakrishnan Ramanathan, Hegde, Jayprasad, Ritesh M. Shah, Pushpak Bhattacharyya, and Sasikumar M. 2008. Simple syntactic and morphological processing can help English-Hindi statistical machine translation. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, pages 171–180.

[Rottmann and Vogel, 2007] Kay Rottmann and Stephan Vogel. 2007. Word reordering in statistical machine translation with a POS-based distortion model. In *Proceedings of the 11th Theoretical and Methodological Issues in Machine Translation (TMI)*, pages 171–180.

[Shen et al., 2008] Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585, Columbus, Ohio, June. Association for Computational Linguistics.

[Simard et al., 2007] Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 508–515, Rochester, New York, April. Association for Computational Linguistics.

[Stolcke et al., 2011] Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. SRILM at sixteen: Update and outlook. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop.*

[Sudoh et al., 2011a] Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Masaaki Nagata, Xianchao Wu, Takuya Matsuzaki, and Jun'ichi Tsujii. 2011a. NTT-UT statistical machine translation in NTCIR-9 PatentMT. In *Proceedings of NTCIR-9*, pages 585–592.

[Sudoh et al., 2011b] Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011b. Post-ordering in statistical machine translation. In *Proceedings of the 13th Machine Translation Summit*, pages 316–323.

[Sudoh et al., 2013] Katsuhito Sudoh, Jun Suzuki, Hajime Tsukada, Masaaki Nagata, Sho Hoshino, and Yusuke Miyao. 2013. NTT-NII statistical machine

translation for NTCIR-10 PatentMT. In *Proceedings of NTCIR-10*, pages 294–300.

[Taku Kudo, 2002] Yuji Matsumoto Taku Kudo. 2002. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69.

[Teh, 2006] Yee Whye Teh. 2006. A bayesian interpretation of interpolated kneser-ney. Technical report, NUS School of Computing Technical Report TRA2/06.

[Tillman, 2004] Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Short Papers*, pages 101–104, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

[Tromble and Eisner, 2009] Roy Tromble and Jason Eisner. 2009. Learning linear ordering problems for better translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1007–1016, Singapore, August. Association for Computational Linguistics.

[Visweswariah et al., 2010] Karthik Visweswariah, Jiri Navratil, Jeffrey Sorensen, Vijil Chenthamarakshan, and Nandakishore Kambhatla. 2010. Syntax based reordering with automatically derived rules for improved statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1119–1127, Beijing, China, August. Coling 2010 Organizing Committee.

[Visweswariah et al., 2011] Karthik Visweswariah, Rajakrishnan Rajkumar, Ankur Gandhe, Ananthakrishnan Ramanathan, and Jiri Navratil. 2011. A word reordering model for improved machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 486–496, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

[Wang et al., 2007] Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 737–745, Prague, Czech Republic, June. Association for Computational Linguistics.

[Wu et al., 2011a] Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011a. Extracting pre-ordering rules from chunk-based

dependency trees for Japanese-to-English translation. In *Proceedings of the 13th Machine Translation Summit*, pages 300–307.

[Wu et al., 2011b] Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011b. Extracting pre-ordering rules from predicate-argument structures. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 29–37, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

[Wu, 1997] Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

[Xia and McCord, 2004] Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of Coling 2004*, pages 508–514, Geneva, Switzerland, Aug 23–Aug 27. COLING.

[Xiong et al., 2006] Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 521–528, Sydney, Australia, July. Association for Computational Linguistics.

[Xiong et al., 2008] Deyi Xiong, Min Zhang, Aiti Aw, and Haizhou Li. 2008. Linguistically annotated BTG for statistical machine translation. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1009–1016, Manchester, UK, August. Coling 2008 Organizing Committee.

[Xiong et al., 2012] Deyi Xiong, Min Zhang, and Haizhou Li. 2012. Modeling the translation of predicate-argument structure for SMT. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 902–911, Jeju Island, Korea, July. Association for Computational Linguistics.

[Xu et al., 2009] Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve SMT for subject-object-verb languages. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 245–253, Boulder, Colorado, June. Association for Computational Linguistics.

[Yamada and Knight, 2001] Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of 39th Annual Meeting*

of the Association for Computational Linguistics, pages 523–530, Toulouse, France, July. Association for Computational Linguistics.

[Yamada and Knight, 2002] Kenji Yamada and Kevin Knight. 2002. A decoder for syntax-based statistical mt. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 303–310, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

[Zens and Ney, 2006] Richard Zens and Hermann Ney. 2006. Discriminative reordering models for statistical machine translation. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 55–63, New York City, June. Association for Computational Linguistics.

[Zens et al., 2004] Richard Zens, Hermann Ney, Taro Watanabe, and Eiichiro Sumita. 2004. Reordering constraints for phrase-based statistical machine translation. In *Proceedings of Coling 2004*, pages 205–211, Geneva, Switzerland, Aug 23–Aug 27. COLING.

[Zhang and Gildea, 2008] Hao Zhang and Daniel Gildea. 2008. Efficient multi-pass decoding for synchronous context free grammars. In *Proceedings of ACL-08: HLT*, pages 209–217, Columbus, Ohio, June. Association for Computational Linguistics.

[Zhang et al., 2008] Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. 2008. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proceedings of ACL-08: HLT*, pages 97–105, Columbus, Ohio, June. Association for Computational Linguistics.

List of Major Publications

[1] Isao Goto, Masao Utiyama, Eiichiro Sumita, Akihiro Tamura, and Sadao Kurohashi. 2014. Distortion model based on word sequence labeling for statistical machine translation. *ACM Transactions on Asian Language Information Processing (TALIP)*, 13(1):2:1–2:21, February.

[2] Isao Goto, Masao Utiyama, and Eiichiro Sumita. 2013b. Post-ordering by parsing with ITG for Japanese-English statistical machine translation. *ACM Transactions on Asian Language Information Processing (TALIP)*, 12(4):17:1–17:22, October.

[3] Isao Goto, Masao Utiyama, Eiichiro Sumita, Akihiro Tamura, and Sadao Kurohashi. 2013d. Distortion model considering rich context for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL2013)*, pages 155–165, Sofia, Bulgaria, August. Association for Computational Linguistics.

[4] Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, Benjamin K. Tsou, Masao Utiyama, and Keiji Yasuda. 2013c. Database of human evaluations of machine translation systems for patent translation. *Journal of Natural Language Processing*, 20(1):27–57, March.

[5] Isao Goto, Masao Utiyama, and Eiichiro Sumita. 2012. Post-ordering by parsing for Japanese-English statistical machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL2012)*, pages 311–316, Jeju Island, Korea, July. Association for Computational Linguistics.

[6] Isao Goto, Masao Utiyama, Takashi Onishi, and Eiichiro Sumita. 2012b. An empirical comparison of parsers in constraining reordering for E-J patent machine translation. *Journal of Information Processing*, 20(4):871–882, October.

[7] Isao Goto, Masao Utiyama, Takashi Onishi, and Eiichiro Sumita. 2011b. A

comparison study of parsers for patent machine translation. In *Proceedings of the 13th Machine Translation Summit*, pages 448–455, September.

[8] Isao Goto and Eiichiro Sumita. 2010. Head- and relation-driven tree-to-tree translation using phrases in a monolingual corpus. In *Proceedings of the 4th International Universal Communication Symposium*, pages 15–22, October.

[9] Isao Goto, Hideki Tanaka, Naoto Kato, Terumasa Ehara, and Noriyoshi Uratani. 2009a. Transliteration using optimal segmentation to partial letters and conversion considering context. *The IEICE Transactions on Information and Systems (Japanese Edition)*, J92-D(6):909–920, June. (in Japanese).

[10] Isao Goto, Naoto Kato, Hideki Tanaka, Terumasa Ehara, and Noriyoshi Uratani. 2006. Automatic acquisition of English equivalents of foreign personal names using the world wide web. *IPSJ Journal*, 47(3):968–979, March. (in Japanese).

List of Other Publications

[1] Isao Goto, Ka Po Chow, Bin Lu, Eiichiro Sumita, and Benjamin K. Tsou. 2013a. Overview of the patent machine translation task at the NTCIR-10 workshop. In *Proceedings of NTCIR-10*, pages 260–286.

[2] Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K. Tsou. 2011. Overview of the patent machine translation task at the NTCIR-9 workshop. In *Proceedings of NTCIR-9*, pages 559–578.

[3] Isao Goto, Kiyotaka Uchimoto, Daisuke Kawahara, and Kentaro Torisawa. 2009b. Automatic acquisition of a bilingual treebank from monolingual web corpora in Japanese and English. *Information Processing Society of Japan, Special Interest Group of Natural Language Processing (IPSJ-SIGNL)*, 192(7):1–8, July. (in Japanese).